October 1979
$2.50

kilobaud

MICROCOMPUTING T.M.

for business . . . education . . . FUN!

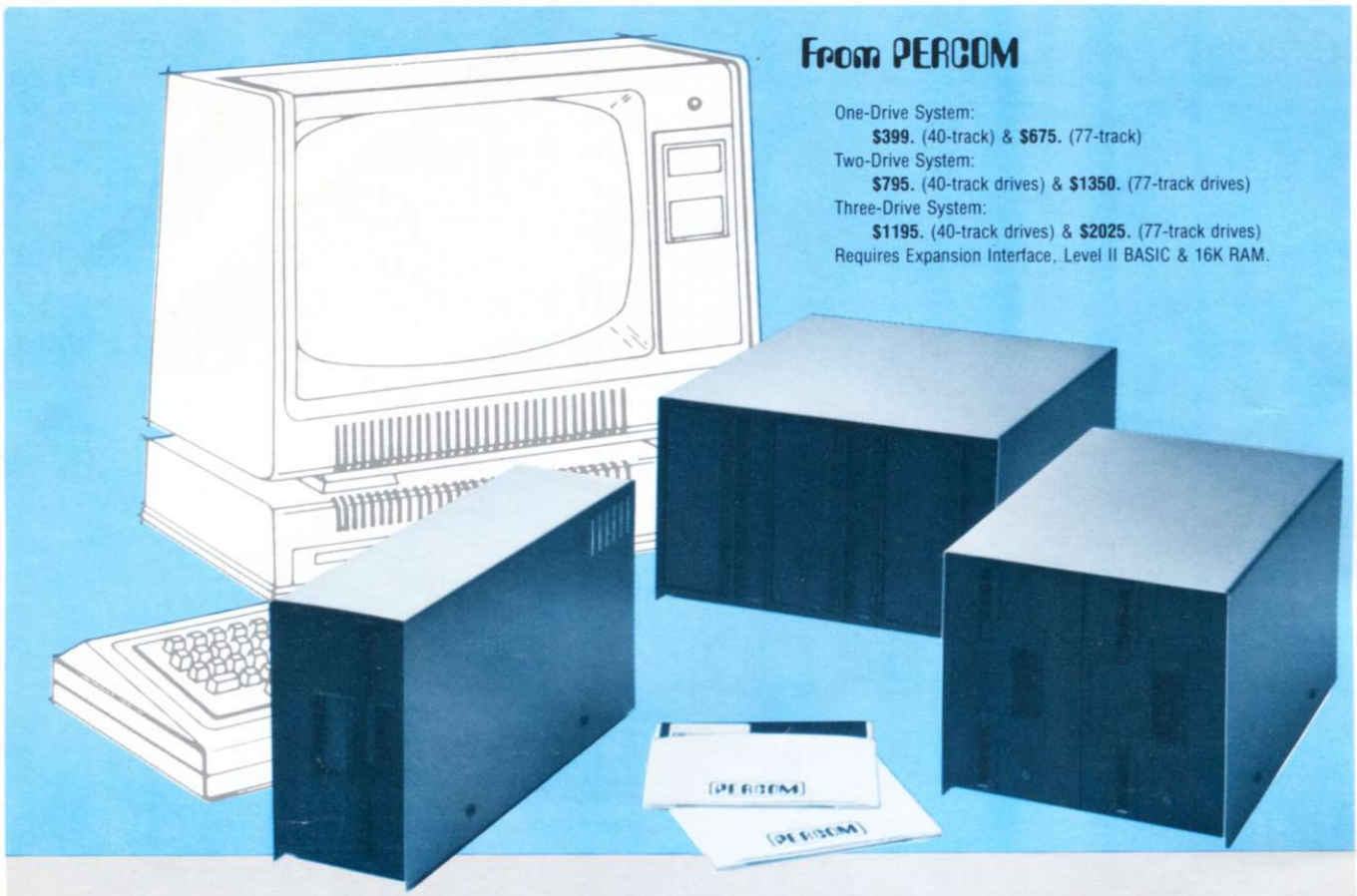
In This Issue—35 Articles

Including:

Special Emphasis on 6800 Systems.....*Six Articles*
The World Power Systems Scam.....*page 56*
Exatron's Stringy Floppy: a Mass Storage Alternative.....*page 98*
Ulysses in Computerland: in Quest of Business Systems.....*page 128*
Inexpensive Logic Analyzer.....*page 172*

Complete Table of Contents on page 5.

1 0
0 74470 65945 0


From PERCOM
One-Drive System:
$399. (40-track) & $675. (77-track)
Two-Drive System:
$795. (40-track drives) & $1350. (77-track drives)
Three-Drive System:
$1195. (40-track drives) & $2025. (77-track drives)
Requires Expansion Interface, Level II BASIC & 16K RAM.

Low Cost Add-On Storage for Your TRS-80*. In the Size You Want.

When you're ready for add-on disk storage, we're ready for you. Ready with six mini-disk storage systems — 102K bytes to 591K bytes of additional *on-line* storage for your TRS-80*.

- **Choose either 40-track TFD-100™ drives or 77-track TFD-200™ drives.**
- **One-, two- and three-drive systems immediately available.**
- **Systems include Percom PATCH PAK #1™, on disk, at no extra charge. PATCH PAK #1™ de-glitches and upgrades TRSDOS* for 40- and 77-track operation.**
- **TFD-100™ drives accommodate "flippy disks." Store 205K bytes per mini-disk.**
- **Low prices. A single-drive TFD-100™ costs just $399. Price includes PATCH PAK #1™ disk.**
- **Enclosures are finished in system-compatible "Tandy-silver" enamel.**

Whether you need a single, 40-track TFD-100™ add-on or a three-drive add-on with 77-track TFD-200™s, you get more data storage for less money from Percom.

Our TFD-100™ drive, for example, lets you store 102.4K bytes of data on one side of a disk — compared to 80K bytes on a TRS-80* mini-disk drive — and 102.4K bytes on the other side, too. Something you can't do with a TRS-80* drive. That's almost 205K bytes per mini-disk.

And the TFD-200™ drives provide 197K bytes of on-line storage per drive — 197K, 394K and 591K bytes for one-, two and three-drive systems.

PATCH PAK #1™, our upgrade program for your TRSDOS*, not only extends TRSDOS* to accommodate 40- and 77-track drives, it enhances TRSDOS* in other ways as well. PATCH PAK #1™ is supplied with each drive system at no additional charge.

The reason you get *more* for less from Percom is simple. Peripherals are not a sideline at Percom. Selling disk systems and other peripherals is our main business — the reason you get more engineering, more reliability and more back up support for less money.

In the Product Development Queue . . . a *printer interface* for using your TRS-80* with any serial printer, and . . . the *Electric Crayon*™ to map your computer memory onto your color TV screen — for games, animated shows, business displays, graphs, etc. Coming PDQ!

™ TFD-100, TFD-200, PATCH PAK and Electric Crayon are trademarks of PERCOM DATA COMPANY.
*TRS-80 and TRSDOS are trademarks of Tandy Corporation and Radio Shack which have no relationship to PERCOM DATA COMPANY.

PERCOM

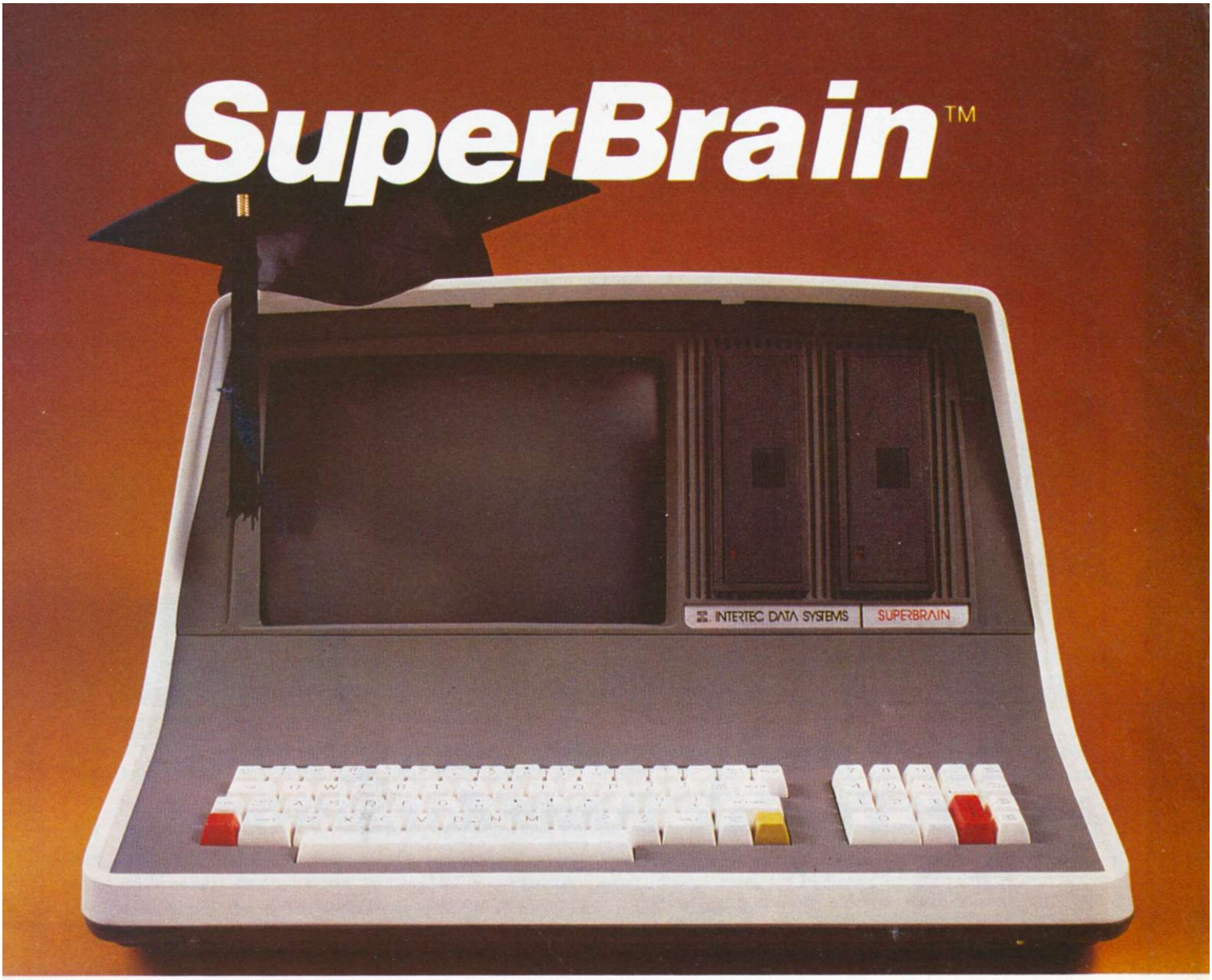
PERCOM DATA COMPANY, INC.
211 N. KIRBY • GARLAND, TX. • 75042

To order add-on mini-disk storage for your TRS-80*, or request additional literature, call Percom's toll-free number: 1-800-527-1592. For detailed Technical information call (214) 272-3421.

Orders may be paid by check or money order, or charged to Visa or Master Charge credit accounts. Texas residents must add 5% sales tax.

Percom *'peripherals for personal computing'*

SuperBrain™



The Honor Graduate

There's been a lot of talk lately about intelligent terminals with small systems capability. And, it's always the same. The systems which make the grade in performance usually flunk the test in price. At least that was the case until the SuperBrain graduated with the highest PPR (Price/Performance Ratio) in the history of the industry.

For less than $3,000*, SuperBrain users get exceptional performance for just a fraction of what they'd expect to pay. Standard features include: two dual-density mini-floppies with 320K bytes of disk storage, 64K of RAM to handle even the most sophisticated programs, a CP/M Disk Operating System with a high-powered text editor, assembler and debugger. And, with SuperBrain's S-100 bus adapter, you can even add a 10 megabyte disk!

More than an intelligent terminal, the SuperBrain outperforms many other systems costing three to five times as much. Endowed with a hefty amount of available software (BASIC, FORTRAN, COBOL), the SuperBrain is ready to take on your toughest assignment. You name it! General Ledger, Accounts Receivable, Payroll, Inventory or Word Processing . . . the SuperBrain handles all of them with ease.

Your operators will praise the SuperBrain's good looks. A full ASCII keyboard with a numeric keypad and function keys. A non-glare, dynamically focused, twelve inch screen. All in an attractive desktop unit weighing less than a standard office typewriter. Sophisticated users will acclaim SuperBrain's twin Z-80 processors which transfer data to the screen at 38 kilobaud! Interfacing a printer or modem is no problem using SuperBrain's RS-232C communications port. But best of all, you won't need a PhD in computer repair to maintain the SuperBrain. Its single board design makes servicing a snap!

So don't be fooled by all the freshman students in the small systems business. Insist on this year's honor graduate . . . the SuperBrain.

*Quantity one. Dealer inquiries invited.

INTERTEC DATA SYSTEMS®

2300 Broad River Road, Columbia, SC 29210
(803) 798-9100 TWX: 810-666-2115

Solve your personal energy crisis. Let VisiCalc™ Power do the work.

With a calculator, pencil and paper you can spend hours planning, projecting, writing, estimating, calculating, revising, erasing and recalculating as you work toward a decision.

Or with VisiCalc and your Apple* II you can explore many more options with a fraction of the time and effort you've spent before.

VisiCalc is a new breed of problem-solving software. Unlike prepackaged software that forces you into a computerized straight jacket, VisiCalc adapts itself to any numerical problem you have. You enter numbers, alphabetic titles and formulas on your keyboard. VisiCalc organizes and displays this information on the screen. You don't have to spend your time programming.

Your energy is better spent using the results than getting them.

Say you're a business manager and want to project your annual sales. Using the calculator, pencil and paper method, you'd lay out 12 months across a sheet and fill in lines and columns of figures on products, outlets, salespeople, etc. You'd calculate by hand the subtotals and summary figures. Then you'd start revising, erasing and recalculating. With VisiCalc, you simply fill in the same figures on an electronic "sheet of paper" and let the computer do the work.

Once your first projection is complete, you're ready to use VisiCalc's unique, powerful recalculation feature. It lets you ask "What if?", examining new options and planning for contingencies. "What if" sales drop 20 percent in March? Just type in the sales figure. VisiCalc instantly updates all other figures affected by March sales.

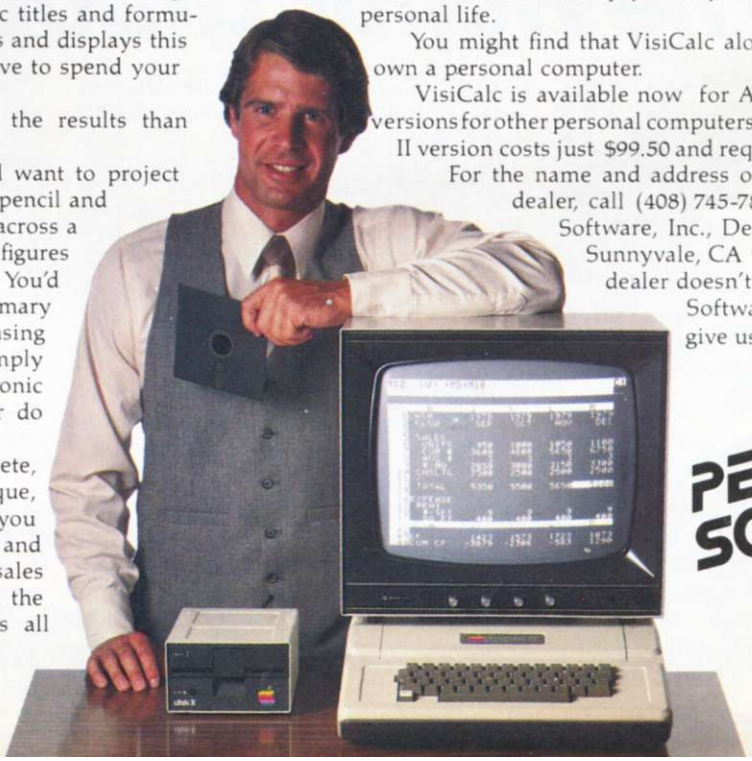
Or say you're an engineer working on a design problem and are wondering "What if that oscillation were damped by another 10 percent?" Or you're working on your family's expenses and wonder "What will happen to our entertainment budget if the heating bill goes up 15 percent this winter?" VisiCalc responds instantly to show you all the consequences of any change.

Once you see VisiCalc in action, you'll think of many more uses for its power. Ask your dealer for a demonstration and discover how VisiCalc can help you in your professional work and personal life.

You might find that VisiCalc alone is reason enough to own a personal computer.

VisiCalc is available now for Apple II computers, with versions for other personal computers coming soon. The Apple II version costs just $99.50 and requires a 32k disk system.

For the name and address of your nearest VisiCalc dealer, call (408) 745-7841 or write to Personal Software, Inc., Dept. K, 592 Weddell Dr., Sunnyvale, CA 94086. If your favorite dealer doesn't already carry Personal Software products, ask him to give us a call.



PERSONAL SOFTWARE

TM—VisiCalc is a trademark of Personal Software, Inc.

*Apple is a registered trademark of Apple Computer, Inc.

kilobaud
MICROCOMPUTING™

contents: oct. '79

30 **The SWTP Computer System** First of a two-part series on disks for SWTP. *P. Shuman, P. Stark*
40 **PAIA 8700 Revisited** Continuing coverage of the 8700. This time—making music. *Rod Hallen*
44 **Inexpensive TRS-80 Printer Interface** Small System Hardware's TRS-232. *Sherman P. Wantz*
46 **Eyes for the AC-30** Mod to add monitoring features to the AC-30. *Jerry Sorrels*
48 **Expanded TRS-80 Disk Operations** First in a two-part series. *Allan J. Domuret*
56 **Anatomy of a Scam** The World Power Systems scam and how it fell apart. *Terry Kepner*
60 **Business Software Made Easy** A business-software-writing primer. *Ernie Brooner*
64 **KIMCTR Measures Capacitance** An enhancement to KIMCTR (May 1979). *Clement S. Pepper*
72 **More TRS-80 Horsepower** Adding 16K of memory and "Level III" BASIC. *Ronald W. Cowart*
78 **Probos V** Build an inexpensive, effective logic probe. *Steve Dominguez*
82 **PET's Keyboard Grows Up** Send your Munchkin-size keyboard to Oz. *Robert W. Baker*
84 **Hurricane!** Track hurricanes with this program. *Bryce Segar*
88 **Video DMA Interface for SWTP Systems** Add direct memory access. *D. Holten, J. Boehme*
94 **Ultimate Consumer Computer** This British-initiated system will be available in the U.S. *Frank J. Derfler*
98 **The Exatron Stringy Floppy** An alternative to cassettes (slow) and disks (expensive). *Bill Harvey*
102 **Calendar Program** What will the boys at the service station say? *Steve Tabler*
104 **Four More Commands for SSB DOS** Increase the BFD-68 repertoire. *Terry Perdue*
110 **Arena** Go into battle with your computer. Richard Price
116 **File Directory Analysis for North Star DOS** Organize your NS disk files. *Edwin Milne*
120 **Report on the Centronics 779 Printer** An "opinionated but accurate" evaluation. *Glen A. Deibert*
122 **Beefing Up PET** Get machine language and BASIC in your PET. *Kendal T. Rogers*
124 **AMI's EVK Series** Part 1 of a three-part series on a 6800 alternative. *David L. Tietz*
128 **Ulysses in Computerland** Obstacles in setting up a business system. *J. Craig McLanahan*
132 **Apple's Documentation Strikes Again** The Apple II Programmer's Aid ROM. *Kenneth Miles*
134 **Caps Lock, Not Shift Lock** Give your shift fingers a break. *Dr. Michael R. Dunlavey*
136 **Hardware Random Number Generator** Try a hardware approach. *Ken Barbier*
138 **Bit Rate Clocks for Your Serial Interface** An alternative to all the UART applications. *R. R. Derynck*
150 **Exploring the Inequality of Bus Buffers** All buffer chips are not created equal. *John R. Cameron*
154 **Elfish Ideas** Speed up your Elf's input-output. *Glen A. Popiel*
158 **Load Programs the SIMPL Way** Check out this author's loader; you might use it too. *Harold B. Combs*
162 **Pig Latin** Is this the ultimate in program protection? *Ken Klosson*
166 **Touch** Handle with care! This icebreaker could be a jawbreaker. *Adrian R. Thornton*
170 **Program Debugging** Straighten out misbehaving programs. *Bob Bosen*
172 **Hardware Testing in Suspended Animation** Build an inexpensive logic analyzer. *K. D. Tentarelli*
176 **Sweetening the Bytesaver** Increase the Bytesaver's usefulness with this mod. *Steven K. Roberts*

Publisher's Remarks—6, Output from Instant Software, Inc.—8, Books—9, PET-pourri—16, New Products—21, Letters—27, Classifieds—108, Contest!—108, Dealer Directory—109, Corrections—109

Cover: This month's cover shows the Exatron Stringy Floppy (ESF) in use as the mass-storage subsystem for the SWTP microcomputer. The ESF fills the gap between the standard cassette and the floppy disk. For less than the cost of the latter, the ESF provides comparable reliability and eliminates the aggravation of slow operating speeds and high error rates of most cassette systems. For more information on the ESF see pages 15 and 98.

micro info

This symbol next to a title in the table of contents indicates that the article is a business-application article.

Manuscripts

Contributions in the form of manuscripts with drawings and/or photographs are welcome and will be considered for possible publication. We can assume no responsibility for loss or damage to any material. Please enclose a self-addressed, stamped envelope with each submission. Payment for the use of any unsolicited material will be made upon acceptance. All contributions should be directed to the *Microcomputing* editorial offices. "How to Write for Microcomputing" guidelines are available upon request.

Editorial Offices:

Pine Street
Peterborough NH 03458
Phone: 603-924-3873, 924-3874

Advertising Offices:

Elm Street
Peterborough NH 03458
Phone: 603-924-7138, 924-7139

Circulation Offices:

Elm Street
Peterborough NH 03458
Phone: 603-924-7296

To subscribe, renew or change an address:

Write to *Microcomputing*, Subscription Department, PO Box 997, Farmingdale NY 11737. For renewals and changes of address, include the address label from your most recent issue of *Microcomputing*. For gift subscriptions, include your name and address as well as those of gift recipients. Postmaster: Send change of address information to *Microcomputing*, PO Box 997, Farmingdale NY 11737.

Subscription problem or question:

Write to *Microcomputing*, Subscription Department, PO Box 997, Farmingdale NY 11737. Please include an address label.

Kilobaud *Microcomputing* (ISSN 0192-4575) is published monthly by 1001001, Inc., Pine St., Peterborough NH 03458. Subscription rates in U.S. are $18 for one year and $45 for three years. In Canada: $20 for one year and $51 for three years. In Europe, send 89,-DM in Euroscheck or send credit card information to: Monika Nedela, Markstr. 3, D-7778 Markdorf, W. Germany. South African Distributor: KB Microcomputing, PO Box 782815, Sandton, South Africa 2146. Australia: For subscriptions write—Katherine Thirkell, Sontron Instruments, 17 Arawatta St., Carnegie, Vic. 3163 Australia. All other foreign subscriptions are $23—one year only (surface mail). Second-class postage paid at Peterborough NH 03458 and at additional mailing offices. Phone: 603-924-3873. Entire contents copyright 1979 by 1001001, Inc. No part of this publication may be reprinted or otherwise reproduced without written permission from the publisher.

PUBLISHER'S REMARKS

Wayne Green

Like to Make $20,000 A Month?

Yes, I know: It sounds interesting if the work isn't too hard and the hours not too long.

There is a need, I feel, for the development of manufacturers' representatives in the microcomputing industry. I was making some calculations of what a rep might be able to earn to see if the field is attractive enough to bring people into this business. The figures look good.

Let's say we divide the country into about 20 areas. With about 1000 computer stores currently in business, this would allocate about 50 stores per area. This is probably all one rep could handle without hiring assistants, since he is going to have to get around to most of these stores at least once a month.

What would the rep do? Well, if we take as an example a rep for Instant Software, he would check the display in a store, make sure it is well placed, make sure it is well stocked, that all the newest releases are prominent, that our posters are on display, that store clerks know how to demonstrate the new programs, answer any questions on programs from clerks, replace any troublesome programs, report to the factory on sales, on any problems, on good vibes and anything else of value to the factory.

As the library of programs increases, a rep is going to have his hands full keeping stores in stock and keeping records of sales, making sure the billing is correct and that stores stay on time with payments. This isn't much of a problem when there are only 100 programs, but Instant Software is aiming at supporting a library of several thousand business, scientific, educational, game and home programs. Every store will have a program library, which will be a large part of their business, probably accounting for much more sales and profit than hardware.

But let's say that our rep is handling just 50 stores, that there are only 500 programs so far released (one year from now at the latest), that the average retail price of the programs is about $15 ($8 for games, $15 for educational programs and $100 for business) and that each store sells an average of . . . how many? Two per month? Three? Five? Ten? Let's take two, so we'll be sure to be underestimating in every way. I think there will be more like 500 stores in each area within a couple years. I'll be disappointed if Instant Software does not have 1000 programs released by mid-1980. But taking the low-ball estimate of 500 programs, two average per month per store, we come up with retail sales per store per month totaling $15,000. Fifty stores will make $750,000 in sales for the rep in his area. A rep would thus get a 10 percent commission on wholesale . . . or about $45,000 per month commissions.

If it turns out that there are 10,000 stores instead of 1000 selling computers, multiply the above figure by 10. If we have twice or three times the number of programs released, double or triple that number. If I'm wrong and sales are only one tenth of my estimate that is still $4500 a *month* in commissions per rep.

If I'm near right in my projections, we're going to have a bunch of millionaire reps.

Stores won't be doing badly either. At the low figure of $15,000 sales per month, their profit will run about $6000! A library of 5000 programs would bring in around $60,000 in *profit* per month!

So how can one get into this lucrative rep business? This is not for the lazy. It means working hard seven days a week for a long time building up the business. It means having a car so you can get to all of the dealers . . . and eventually a van or even truck for delivering programs and promotion materials. One should have a familiarity with microcomputers, with selling, with bookkeeping, with software, plus the determination to succeed.

It will take some time to develop a reputation and to build up commissions. Of the 50 stores in an area, probably no more than five or ten are already stocking Instant Software . . . the rest will have to be sold. There will be resistance . . . there always is to something new, no matter how beneficial it is. Some dealers won't want to carry software for the TRS-80 because they aren't selling it. This means you'll have to show them how this will bring more customers into their store and permit them to take advantage of the rapidly growing and highly lucrative market in accessories for the TRS. It means showing the dealer that most newcomers to computing are coming in via their purchase of a TRS or other small system rather than as a hobbyist, as it was a couple years ago.

But even with five dealers and only 100 program packages and with the average retail price at $8, a rep can right now, without growth, make nearly $500 per month in commissions. Each extra store brought on line should increase the monthly commissions by a minimum of $160.

If we add in the 2300 Radio Shack associate stores and franchise stores, all of which have the freedom to deal directly with Instant Software through a rep firm, we're adding about 115 more stores per rep area . . . right now. Can you sell TRS programs to Radio Shack stores? If you can't, then you sure should stay in some other business. Better go get a degree as an engineer and not have to work again.

Paris Micro/Expo

Sybex's micro show in Paris was well attended, with some very familiar names exhibiting. Here are some more photos of people and exhibits at the show.

Software Support Critical

Several of the smaller manufacturers of microcomputer systems haven't thought very far ahead and as a result are not aware of the importance of software to their sales when the time comes that other systems have a large supply of inexpensive software.

Let's exaggerate just a trifle to get the concept. Let's suppose that a prospective microcomputer customer has been lured into a store and is looking over system A vs system B. One has the support of several thousand programs—some right down his alley for business, some just a help in his business, plus some games, security, data base, word processing, etc., which look good. The other system has no programs available at all. The price on the two systems is similar. Which do you think he will buy?

Here is the Heath exhibit. Actually, the interest in kits seemed to be less than expected, so the new Heath line of assembled units will probably better fit the European temperament. Europeans don't seem to be interested to a high degree in the hobby aspect of microcomputers, but rather in using them for practical business and scientific applications.

One of the most crowded booths all through the show was the Tandy booth. The TRS-80 is a best seller in some parts of Europe, and just getting started in others.

Sybex did a brisk business in their books, some of which have now been translated into German and French.

Reinhard Nedela and Sherry were busy at the Kilobaud Microcomputing *booth with subscriptions and Instant Software sales.*

Actually, the prices of the two systems will not be similar because the software support of one system will so increase its sales that there will be substantial cost reductions to pass along to the customer . . . and to the dealer in terms of higher profits. A doubling of production in this field seems to result in a dropping of manufacturing costs of about 25 percent. This can mean a drop of about four times that to the customer.

Unfortunately for the systems manufacturer who does not push hard for software support, the more software developed for another system, the more will be developed. It is self-generating. And the more software there is, the better all of it is going to be. Instant Software is getting much more selective about the software being accepted for publication. With some dismay, we are already seeing our rejects coming on the market from smaller publishers, and we know that disappointment is ahead for the people who get suckered into buying these programs, which are usually published at a higher price than the better programs ISI is publishing.

The growth of software publishing has been held back to some degree by the costs of lab hardware. The more money that has to be put into hardware, the less is available for salaries and the people needed to get the work done. Manufacturers may eventually recognize this and work out some system for helping software publishers with their work . . . and with equipment.

Business Software

The development of top-notch business software for microcomputers is, as I predicted, taking a lot longer than games . . . and for obvious reasons for anyone who gives it any thought. It takes a lot of time, experience and genius to write really good business programs; plus you have to have both the hardware available and the language and operating system before you can get a good start.

Much of the business software now coming into Instant Software for publication has been developed by individual programmers for custom applications . . . often in conjunction with a local Radio Shack store. Some of these programs are indicative of what we can expect in the future—programs, for instance, for specific hospital functions, for working with blood testing systems to speed up their operation and print out the results of hundreds of tests, to work with patient monitoring equipment to keep a record of many factors and alert someone if anything gets out of acceptable parameters, to determine the doses of certain medicines from tests by other equipment.

We'll be seeing this depth of business software developing in many fields. We've already some surprisingly sophisticated real

Bob Jones of Interface Age *had a booth too. The airlines lost his luggage for a couple of days, making the first part of the visit a bit hectic.*

Though Harold Buchbinder, the publisher of Personal Computing *was scheduled to talk, he never showed up, and the PC booth was vacant.*

kilobaud MICROCOMPUTING™

PUBLISHER/EDITOR
Wayne Green

EXECUTIVE VICE PRESIDENT
Sherry Smythe

ASSISTANT PUBLISHER/EDITOR
Jeffrey D. DeTray

MANAGING EDITOR
John Barry

EDITORIAL ASSISTANTS
Dennis Brisson
Susan Gross

ADMINISTRATIVE ASSISTANT
Dotty Gibson

PRODUCTION DEPARTMENT
MANAGER:
Noel R. Self
STAFF:
Cathy Callahan
Robert Drew
James H. Gray II
Bruce Hedin
Carl Jackson
Ken Jackson
Dion Owens
Patrice Scribner
Robin M. Sloan
John W. White

TYPESETTING
Barbara J. Latti
Sandie Gunseth
Mary Kinzel
Holly Walsh

PHOTOGRAPHY
W. H. Heydolph
Tedd Cluff
Terrie Anderson

PROJECTS EDITOR
Jim Perry

ASSOCIATE EDITORS
Phil Feldman
Rod Hallen
Len Lindsay
Tom Rugg
Peter Stark
Sherm Wantz
Ken Widelitz

BOOKKEEPER
Knud E. M. Keller

MARKETING/CIRCULATION MGR.
Robert LaPointe

BULK SALES MGR.
Judy Waterman

ASST. CIRCULATION MGR.
Donna Taylor

ASST. MARKETING MGR.
Harold Stevens

CIRCULATION
Pauline Johnstone

COMPUTER PROGRAMMING
Richard Dykema

INSTANT SOFTWARE MICRO LAB
William F. Gollan
Scott Spangenberg
Palmer T. Wolf
Stan Wszola

EUROPEAN MARKETING DIR.
Reinhard Nedela

AUSTRALIAN DISTRIBUTOR
Katherine Thirkell

ADVERTISING
Aline Coutu, Mgr.
Bill York
Marcia Stone
Nancy Ciampa
Gaye Halbig
Lori Mugford
Rita Rivard
Eve Anderson
Kevin Rushalko
Carol Symonowicz

estate packages, but they are nothing to what we'll have in a year or two when most brokers will not only have immediate information on their inventory, but also data on all neighboring inventories and a wide range of calculations possible on the myriad of methods for financing real estate. The forms will be done automatically, along with all billing, title search and legal aspects.

I had to laugh at the innocence of one manufacturer who thinks of microcomputer business software as a "bad joke . . . written by amateurs." I hope he keeps his non-micro business going . . . he may need it. Many of the programs we are seeing are being written by old-time professional programmers, some with PhDs in computer science. Being a PhD doesn't mean you are stupid and don't know where the money is, and the big money for the freelance computerist lies in software—business software right now, then educational, and finally all kinds. Computers can handle just about any timesaving application once the programs are available. I will again predict that the eventual uses for our microcomputers will surprise all of us. The surface is just being scratched and, as new programs are developed, the need for more will follow.

As a publisher, I can look forward to word processing helping prepare material for publication. Computers can then be used to lay out pages and set them ready for printing, even to making the printing plates directly. Editors can use a computerized index to list published material for fast referencing. Computerized soliciting of bids for printing and paper will provide efficiencies and cut costs. Computerized mail will speed material for publication from remote writers and editors. Computers can help with determining the imposition of pages on a larger press. Computers help work out the best ratio of advertising to editorial material, taking into consideration the number and placement of two-color ads, three-color ads, four-color ads and editorial material with color.

Then we need computers to take care of daily orders from individuals for current and back issues. Computers can also handle orders from dealers and wholesalers. Computers will speed up shipping and billing, bookkeeping, handling plastic money, keeping an inventory record and sales records of all transactions. These, in turn, will provide reports for the circulation people so they know which ads are working best and what covers sell better than others. Add in the replies to various polls, and you have more need for computers to provide a better product at a lower price.

The microcomputer will be able to help every business to operate at lower cost and with more efficiency. Plus it will take a lot of the drudgery out of the work by doing most of the routine. I think we will be seeing all of these programs evolving for microcomputers, and I'm hoping that Instant Software will be the major distributor of these programs . . . all over the world.

In May we opened up Europe for software distribution. South Africa should be next in September. Asia will be opened up in October. The more outlets for software there are, the more royalties for the programmers . . . and the better programs we'll have to help sell systems. Will systems with poor software support sell? I can't think of any reason they should.

New Apple!

Though I've seen no releases yet from Apple, the stores are now selling a new model, the Apple II+. We'll probably hear more about the changes involved, but the immediate difference has to do with the BASIC ROM, which comes built in.

The Apple II came with Integer BASIC installed, and you had to add Applesoft via a plug-in board to get floating point numbers, high-resolution routines and a much more complete BASIC. The Apple II+ system has the Applesoft built in, and you have to plug in an Integer BASIC card to run your programs written in that version . . . or to speed up some functions. Delivery is expected soon on the Integer BASIC boards.

The Instant Software Apple programs—Golf (0018A), Bowling (0040A), Math I (0073A), Math II (0098A)—are mostly in Applesoft. Trilogy (0040A) and a second run of Math I (0040A) are in Integer BASIC. Apple owners not having both languages should be careful when buying packaged programs and make sure they match the ROM they're using.

Ripped Off?

If you have a serious problem with a microcomputer firm, send them a letter with *all* the facts in detail, plainly and simply, and send a copy to me. *Kilobaud Microcomputing* protects its readers more than any other magazine.

OUTPUT FROM ISI

Sherry Smythe

The number of programs available for microcomputers is growing slowly; most of the new entries still are games. Dealers are not enthusiastic about this. They prefer to see business-oriented software since these programs will not only provide program sales, but also will help to sell their systems.

Oddly enough, one of the best business programs released (Business Cycle Analysis, a powerful management tool) so far has been for the TRS-80, and Radio Shack stores are prohibited from selling programs other than those released from Ft. Worth. This may be costing stores thousands of computer sales and costing Radio Shack into the millions. Meanwhile, computer stores are enjoying the bonanza, and TRS-80 owners are forced to find their neighborhood computer store to get any serious support for their system.

Many dealers have been reluctant to stock programs for systems other than those they are selling. The dealers who are stocking programs for all systems are finding that their program sales are substantially better as a result. Perhaps 85 percent of the newcomers to microcomputing these days are entering via a TRS-80, and stores with no TRS-80 products are effectively chasing away their best sales prospects.

Prospects for business programs are better and better. About half the programs now being submitted for publication by Instant Software are business-oriented. Oh, the umpteenth financing program arrived long ago and aroused as much interest as another version of Hangman or Mastermind. When you consider how many business programs there are to be written, you wonder at this reinventing of the wheel. Many programmers are now getting together with business friends and developing program packages for them for mass publication.

The crew at Instant Software has been growing. The programs are now all cataloged on a TRS-80 with a couple disks, as are the associate editors and their specialties. There is more and more work for associate editors as more programs are being translated for use on more systems and also into foreign languages . . . in addition to the usual evaluations of newly submitted programs.

Disks

Increasing numbers of programs are being submitted on disks, which is fine with us. There is little reason for a business, even a small business, not to invest in at least a dual-disk system. Without that minimum of on-tap memory, few significant programs can be used. Should the day come when it seems better to sell a program on a disk, Instant Software will go that route. In the meantime, most programs—even the disk-based programs—are easily transportable via cassettes, and this saves some money.

We're geared up to check out and produce programs based on NEWDOS, which is getting very good marks from users. Judging from reader reports, it appears to be superior to TRSDOS.

Educational Programs

Games are *not* needed. Programs already listed as available are not badly needed. Educational programs are very much in demand. But look here—if you are going to write a program to teach

(see OUTPUT, page 190)

BOOK REVIEWS

The Cheap Video Cookbook
Don Lancaster
Howard W. Sams & Co., Inc.
Indianapolis IN
1978, 256 pp., paperback

All of us at one time or another have probably yearned for a video terminal. Some of us were rich enough to afford one, but the rest of us had to be content to struggle with our eight puny data LEDs until the day came when we could afford a good terminal for our own. Well, that day has arrived. Don Lancaster has come to our rescue with his latest "cookbook," *The Cheap Video Cookbook*, which explains in detail how you can design and build your own terminal for less than $20!

The big secret behind cheap video is that there is relatively little hardware used—a handful of gates and a character generator. The key to the system is the microprocessor. Utilizing several routines in conjunction with a video interface, the microprocessor does all the "housework" such as generating the video and sync signals. By changing your software structure, you can change your terminal's characteristics. As a result, you can have a very simple terminal or a very versatile one with almost any feature you could want. Features include: any display format from 1 × 16 to 24 × 80; various character fonts from 5 × 7 dot matrix to Old English; 256 × 256 black and white graphics; 96 × 128 color graphics; a host of control features such as scrolling and reversed video.

The important thing to notice is that no matter how advanced a terminal you want, the price you pay for an extra option is not another fancy IC, but rather another portion of your memory. And you don't need much memory either; 1K will keep you busy for quite a while.

While the video system described in this book has a lot going for it (low cost and versatility), it also has a few disadvantages. The most serious one, perhaps, is that it will not readily work with many CPUs. The CPU must meet certain demands if the video interface is to function properly. One requirement is that the microprocessor be able to advance the program counter once every microsecond. It must also have 16 always-present and fully decoded address lines. Some systems just cannot meet these requirements without additional hardware, which means more $$$.

Since the video interface described in the book was designed for 6500- and 6800-based systems, computers with such chips as the F-8, SC/MP, 1802 and even the 8080 will require modification to be compatible. Depending on the system, these modifications could be extensive and difficult. Not only that, but the software will have to be translated as well. In short, conversion to another CPU will introduce new problems and take your creative talents to solve.

Since the video system in this book is based around the 650x microprocessors, it is necessary that you be familiar with the 6500 machines as well as the KIM-1. The author recommends that prior to this book you read the following: *The TV Typewriter Cookbook* (Sams 21313); *An Introduction to Microcomputers*, Volumes I & II (Osborne 2001 and 3001, respectively); user's manuals for the 6500 and KIM-1; user's manuals for your system.

Obviously, *The Cheap Video Cookbook* is not intended for the beginning hobbyist.

The Cheap Video Cookbook contains five chapters. Chapter 1 deals with the foundations of cheap video, how it works, what it can do and which CPU to use. Two new and very important concepts are also introduced: the Scan Microinstruction and the Upstream Tap.

Chapter 2 shows how to design the software for the cheap video system. Here we discover how to build programs for the scan microinstruction, alphanumeric displays, high-resolution graphics and full-performance cursors. Editors and graphics loaders are also described in detail. Incidentally, owners of KIM-1 systems will be able to use these programs right out of the book.

Chapter 3 involves the design of the video interface. The circuit is designed step-by-step from the instruction decoding PROMs to the output circuitry. Finally, you are shown how to interface the system with a KIM-1 and how to add a video input to an ordinary TV set.

Chapter 4 is like an instruction booklet for the many electronic kits produced these days. The first part of this chapter discusses circuit theory and the construction of Don Lancaster's latest TVT, the TVT 6-5/8. All components are installed one by one, and each step is checked off after completion. There are also several excellent full-size patterns for the PC boards and solder mask. In addition, chapter 4 contains complete construction details for four additional add-on modules that give the TVT 6-5/8 extra features such as graphics (black and white and color) and lowercase alphanumerics. With the TVT 6-5/8 and each module is a troubleshooting guide to help isolate errors in construction and programming. The chapter winds down with information concerning modifications and part of a chess program utilizing the color-graphics module.

The principles of transparency, which allow the CPU to compute data and run the video interface simultaneously, are discussed in chapter 5. Several methods are shown along with the 6500 software required to make them work.

Four appendices finish the book. Appendix A is simply a listing of the ASCII code, while appendix B is an octal-hex-decimal conversion chart (rather handy!). Appendix C contains the pin-outs of several ICs used throughout the circuits described in this book. Appendix D repeats the PC board patterns scattered in chapter 4.

Overall, the book appealed to me. The material is presented in a clear, easy-to-understand manner (provided, of course, you have the necessary background material previously mentioned) and can be mastered in a fairly short time. My biggest complaint is that the software and most of the hardware is designed primarily for the 6500 series CPUs.

Although I believe that every computer system can be adapted to use cheap video techniques in some way, I don't recommend this book to anyone who doesn't own a 6500 or 6800 microcomputer. However, if you do own another type of system and are sincerely interested in a $20 video terminal, and you are willing to put much time and effort into it, then I suggest you either borrow a copy or read the first chapter in the bookstore before you purchase the book. That way you can determine if cheap video is for you without unnecessarily investing money that would have been better spent on some more RAM or even another book. If you opt for the latter method, I suggest you hide behind a bookshelf to avoid evil looks from the shopkeeper!

Steve Dominguez
Golden CO

Programming in PASCAL
Peter Grogono
Addison-Wesley, Inc.
1978, $10.95

If you've been wondering what all the hoopla about PASCAL is then this is the book for you. Since you'll probably be seeing a lot of PASCAL anyway, what with the Apple II, TRS-80 and other versions coming out, you may as well buy the book now and save yourself some time later.

You can probably already tell that I like this book. As a matter of fact, I can't say enough good things about it. If we'd had textbooks like this when I was in school, we wouldn't have needed teachers. Grogono takes you by the hand and leads you through the very most fundamental concepts of programming through the most powerful features of PASCAL. The text is liberally illustrated with actual programs and easy-to-understand syntax diagrams (finally—easy-to-read syntax diagrams!) that make learning the language a snap.

Chapter 1, an introduction to computer programming, includes a discussion of what a program is and a short introduction to PASCAL to help you get your feet wet. I have a strong feeling that even a non-programmer can learn to write PASCAL programs with no help but this book. Chapter 1 is part of the reason for this.

Chapters 2, 3 and 4 cover much of the language. With these first four chapters under your belt you'll be ready to try some programs of your own. The clarity of the book and the well-chosen examples should make these about a weekend's worth of reading even for a total beginner.

Chapter 5 covers some of the functions PASCAL performs,

(see REVIEWS, page 14)

OHIO SCIENTIFIC'S ⌐013
SMALL SYSTEMS JOURNAL

Introduction

Our objective in this issue is to give the reader brief descriptions of two systems included in the OS-DMS grouping—General Ledger and Payroll—which have not yet been shown in our previous articles. Several of the reports created by these systems will be shown in order that the reader might better understand the purposes of the systems.

OS-DMS General Ledger System

In the normal office environment there is a need to keep track of how much business is done by the firm. Whether the amounts earned are normal or not and whether the company is making a profitable return on its investments are samples of questions constantly posed about a business. The figures utilized to answer these questions are measurements of money, and the process of working with these figures is called accounting. The most basic accounting report used in the process of keeping track of a firm's activities is the general ledger, and any specific series of operations designed to reveal the financial status of a firm is called a general ledger system.

A general ledger system may be manual or automated. It may be very simple for a small firm or extremely complex for a large corporation with multiple divisions or departments. In all cases, however, the basic objectives of having a general ledger are the same: enable management to know how the firm is faring financially as well as make decisions and forecasts from an informed position.

The OS-DMS General Ledger System is designed to fit into this office environment in the following manner.

Cash is earned or borrowed, spent or loaned, and in other ways comes and goes. This cash flow should be carefully controlled. This is the objective of the cash receipts and disbursements portion of this system. All incoming and outgoing funds are entered in detail on the computer through the console terminal in a clerical operation. The payer or payee name, the date, a reference number of some type, a general ledger account number and the dollar amount of each activity are put into the computer as the transactions take place. An input journal of the events that take place in this operation is printed so that the person overseeing the cash flow (the manager) may review the activities when the need arises. Periodically, cash receipts journals and cash disbursements journals are printed in order to have records of the cash activities of the firm. At the ends of accounting periods, totals from each general ledger account affected by the cash flow from the period are posted to the general ledger, and the cash books for that period are closed.

While cash transactions make up a large activity area in business, many accounting entries do not include a movement of funds. Buildings and equipment must be depreciated. Items are bought and sold for promises to pay. Taxes and other expenses are accrued through a certain date. All these entries must appear in the company's books.

In this general ledger system, these types of transactions are entered separately from the cash movements. The routine through which these entries are made prints the general journal, a normal accounting entry ledger. Through this general journal, records are maintained as to what activity has gone on at what time.

The data entry is constant in both the cash and non-cash areas. The end of an accounting period is a point at which the entries for one measurable time period—usually a month or a year—are stopped so that their contents may be worked with, totaled, compared to other like periods and evaluated. This is the closing of the books. Reports such as the general ledger, the operating statement and the balance sheet are printed for middle and upper management's needs. Data is stored for historical reference and analysis. Management personnel now have reports concerning a standard time period in a common format with which they may make decisions that affect the future of the firm.

This system produces all the reports mentioned above. It also takes care of expansion or restructure of the general ledger system. Protection of the data through disk backup is an integral part of this operation.

The OS-DMS general ledger is open-ended. Though this system is large, and in itself is complete, it is just a part of the needed product for most users. Through the use of the various other modules and the nucleus utilities of OS-DMS, there is much that may be done to expand its capabilities. Every user's needs are different, so this is a necessary state. It is felt that the base provided here can support this expansion.

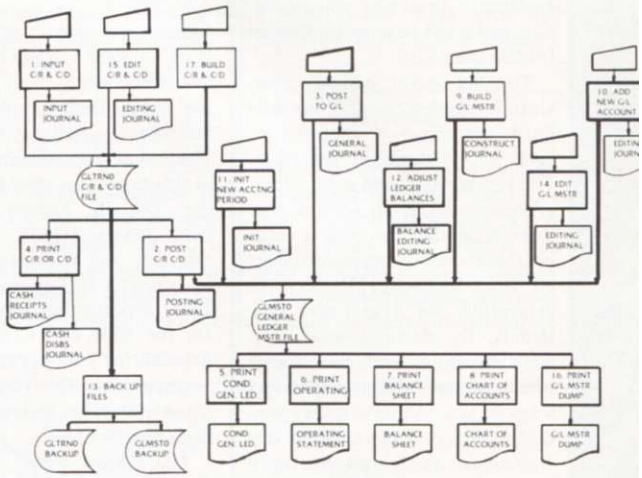


FIGURE 1. OS-DMS GENERAL LEDGER LOGIC FLOW DIAGRAM

The first area we shall look at is the input function. This general ledger system has tw specific input routines—one to handle cash receipts and disbursements and the other work with all other types of entries. The main difference between the two is that the form creates a detailed transaction file of all entries, whereas the latter posts directly to t general ledger master file. Sample input journals are as follows:

```
CASH RECEIPTS OR DISBURSEMENTS INPUT JOURNAL          01/26/79      9:
==========================================================================
  PAYER/PAYEE                 DATE        REF #      ACCT #     AMOUN
STARTING NEW ACCOUNTING PERIOD.
SUBTOTAL: 0.00
==========================
ADD RECORDS
••• CASH RECEIPTS •••
  JOHNSON BAKERY              01/03/79    203756     1020       250.0
  SMITH LUMBER CO             01/05/79    33398      4010       159.9
  APEX STORES INC             01/10/79    12205      1020       500.0
  CONSOLIDATED TRUST          01/10/79    330856     4010       3579.5
  JACKSON DRUGS               01/12/79    222556     1020       300.0
  W & W CAFETERIA             01/15/79    330564     1020       325.5
  CITY FINANCE CO             01/17/79    440322     4010       1399.5
SUBTOTAL: 6514.51
==========================
ADD RECORDS
••• CASH DISBURSEMENTS •••
  FIRST NATIONAL BANK         01/10/79    44032      2015       115.0
  WILLIAMS EQUIPMENT          01/12/79    22034      5010       304.7
  ACME INSURANCE CO           01/15/79    33944      5250       157.0
  OFFICE PAYROLL              01/15/79    22045      5410       3022.5
SUBTOTAL: 3599.29
==========================
••• END RUN •••
```

FIGURE 2. CASH RECEIPTS OR DISBURSEMENTS INPUT JOURNAL

GENERAL JOURNAL 1/23/79 9:46

DATE	ACCT	NAME/DESCRIPTION	REF NUMBER	DEBITS	CREDITS
1/10/79	1030	JOHNSON EQUIPMENT	33085	2579.40	
1/10/79	2010	ACCOUNTS PAYABLE	33085		2579.40
1/10/79	1040	HAYES OFFICE SUPS	33057	13.78	
1/10/79	2010	ACCOUNTS PAYABLE	33057		13.78
1/13/79	1330	GLOVER SUPPLY CO	33058	308.00	
1/13/79	2010	ACCOUNTS PAYABLE	33058		308.00
1/16/79	4010	ALLIED STORES	55098	4500.00	
1/16/79	1020	ACCOUNTS RECEIV.	55098		4500.00
1/16/79	1020	55098 CORRECTION	55098-C	4500.00	
1/16/79	4010	55098 CORRECTION	55098-C		4500.00
1/18/79	1030	ALLIED STORES	55098	4500.00	
1/18/79	4010	SALES	55098		4500.00
TOTALS:				16401.18	16401.18
NET:				0.00	

FIGURE 3. INPUT GENERAL JOURNAL

In addition to the printouts from these input runs, we should also look at the end-of-period statements. The first of these is the balance sheet, and the standard OS-DMS format is as follows:

BALANCE SHEET
FOR PERIOD ENDING 9/30/78

ASSETS			
CURRENT ASSETS			
CASH	8591.99		
ACCOUNTS RECEIVABLE	6572.76		
INVENTORY	4502.78		
PETTY CASH	753.83		
TOTAL CURR ASSETS		20421.36	
FIXED ASSETS			
LAND	12301.38		
BUILDINGS	51348.55		
DEPRECIATION—BUILDINGS	-9417.67		
EQUIPMENT	10753.20		
DEPRECIATION—EQUIPMENT	-4301.75		
TOTAL FIXED ASSETS		60683.71	
OTHER ASSETS			
PREPAID DEPOSITS	528.56		
TOTAL OTHER ASSETS		528.56	
TOTAL ASSETS			81633.63
LIABILITIES			
CURRENT LIABILITIES			
ACCOUNTS PAYABLE	3419.82		
DIVIDENDS PAYABLE	4106.09		
INCOME TAX PAYABLE	1821.69		
INTEREST PAYABLE	532.43		
INSURANCE PAYABLE	378.66		
TOTAL CURR LIABILITIES		10258.69	
LONG-TERM LIABILITIES			
LONG-TERM NOTE	9721.45		
TOTAL LONG-TERM LIABS		9721.45	
OTHER LIABILITIES			
PREPAID SERVICES	2670.18		
TOTAL OTHER LIABILITIES		2670.18	
TOTAL LIABILITIES			22650.32
STOCKHOLDERS' EQUITY			
COMMON STOCK	40000.00		
RETAINED EARNINGS	18983.31		
TOTAL STKHOLDERS' EQUITY			58983.31
TOT LIABILITIES & EQUITY			81633.63

FIGURE 4. BALANCE SHEET

Along with the balance sheet, the other primary end-of-period report is the operating statement, often referred to as a profit and loss statement. The standard OS-DMS operating statement has the following format:

OPERATING STATEMENT
FOR PERIOD ENDING 9/30/78

DESCRIPTION	CURR AMOUNT	% CURR SLS	YTD AMOUNT	% YTD SLS
REVENUES				
SALES	18509.10	101.06	167705.43	101.29
SALES RETURNS	194.35	1.06	2142.56	1.29
NET SALES	18314.75	100.00	165562.87	100.00
COST OF SALES				
BEGINNING INVENTORY	1508.30	8.24	2503.48	1.51
ADD: PURCHASES	10210.44	55.75	103750.54	62.67
DED: PURCH RETS & ALLOWS	820.75	4.48	9331.25	5.64
TOTAL GOOD AVAIL	12539.49	68.47	115585.27	69.81
DEDUCT: ENDING INVENTORY	4502.78	24.59	4502.78	2.72
TOTAL COST OF SALES	8036.71	43.88	111082.49	67.09
GROSS PROFIT	10278.04	56.12	54480.38	32.91
OPERATING EXPENSES				
SALES EXPENSES				
SALES SALARIES	2053.69	11.21	19858.55	11.99
ADVERTISING	359.98	1.97	3476.18	2.10
TOTAL SALES EXPENSES	2413.67	13.18	23334.73	14.09
GENERAL EXPENSES				
SALARIES	684.30	3.74	6615.50	4.00
RENT	152.55	.83	1476.21	.89
TELEPHONE	392.23	2.14	4070.63	2.46
TOTAL GENERAL EXPENSES	1229.08	6.71	12162.34	7.35
TOTAL OPERATING EXPENSES	3642.75	19.89	35497.07	21.44
NET PROFIT—LOSS	6635.29	36.23	18983.31	11.47

FIGURE 5. OPERATING STATEMENT

Other reports necessary for the smooth operation of a general ledger system are the utility reports. Included in this category are printouts such as a chart of accounts and a formatted master file listing. In the OS-DMS system, these two reports appear as follows:

CHART OF ACCOUNTS 9/30/78

ACCT #	DESCRIPTION	TYPE RCD	DEBIT/CREDIT
	ASSETS		
	CURRENT ASSETS		
1020	CASH	ASSET	DEBIT
1030	ACCOUNTS RECEIV	ASSET	DEBIT
1040	INVENTORY	ASSET	DEBIT
1050	PETTY CASH	ASSET	DEBIT
	TOTAL CURR ASSETS	MINOR TOTAL	
	FIXED ASSETS		
1510	LAND	ASSET	DEBIT
1520	BUILDINGS	ASSET	DEBIT
1530	DEPRE—BUILDINGS	ASSET	CREDIT
1540	EQUIPMENT	ASSET	DEBIT
1550	DEPRE—EQUIPMENT	ASSET	CREDIT
	TOTAL FIXED ASSETS	MINOR TOTAL	
	OTHER ASSETS		
1810	PREPAID DEPOSITS	ASSET	DEBIT
	TOTAL OTHER ASSETS	MINOR TOTAL	
	TOTAL ASSETS	MAJOR TOTAL	
	LIABILITIES		
	CURRENT LIABILITIES		
2020	ACCOUNTS PAYABLE	LIABILITY	CREDIT
2030	DIVIDENDS PAYABLE	LIABILITY	CREDIT
2040	INCOME TAX PAY	LIABILITY	CREDIT
2050	INTEREST PAYABLE	LIABILITY	CREDIT
2060	INSURANCE PAYABLE	LIABILITY	CREDIT

FIGURE 6. PORTION OF CHART OF ACCOUNTS

OS-DMS GENERAL LEDGER MASTER FILE LISTING 9/30/78

ACCT #	DESCRIPTION	PREV BAL	TRNSACTNS
1000	ASSETS		
1010	CURRENT ASSETS		
1020	CASH	6270.46	2321.53
1030	ACCOUNTS RECEIVABLE	7191.22	-618.46
1040	INVENTORY	2503.48	1508.30
1050	PETTY CASH	778.40	-24.57
1499	TOTAL CURR ASSETS		
1500	FIXED ASSETS		
1510	LAND	12301.38	0.00
1520	BUILDINGS	51348.55	0.00
1530	DEPRECIATION—BUILDINGS	-8692.23	-725.44
1540	EQUIPMENT	10753.20	0.00
1550	DEPRECIATION—EQUIPMENT	-3980.66	-321.09
1799	TOTAL FIXED ASSETS		
1800	OTHER ASSETS		
1810	PREPAID DEPOSITS	507.21	21.35

CURR BAL	RCD TYPE	DEBIT/CREDIT	SOURCE	OPEN 1	OPEN 2
	H (HEADER)				
	H (HEADER)				
8591.99	A (ASSET)	D	3W	0	CASH
6572.76	A (ASSET)	D	4E	0	A/R
4502.78	A (ASSET)	D	25	0	INV
753.83	A (ASSET)	D	7T	0	0
	3 (MINOR TOT)				
	H (HEADER)				
12301.38	A (ASSET)	D	M	0	0
51348.55	A (ASSET)	D	M	0	0
-9417.67	A (ASSET)	C	M	0	0
10753.20	A (ASSET)	D	M	0	0
-4301.75	A (ASSET)	C	M	0	0
	3 (MINOR TOT)				
	H (HEADER)				
528.56	A (ASSET)	D	M	0	0

FIGURE 7. PART OF GENERAL LEDGER MASTER FILE LISTING

OHIO SCIENTIFIC 1333 S. Chillicothe Road · Aurora, Ohio 44202 · (216) 562-3101

Reader Service—see page 227

In addition to the reports shown above, of course, there is some form of printed documentation created by practically every operation performed within the system. This is designed to assist the user in recording his actions, but even more, to allow him to see where he has skipped an operation. In all cases, the objective of the system has been to assist the user in the management of his firm and give him better control.

OS-DMS Personnel/Payroll System

Payroll is a normal part of any business that employs more than one person. In order to comply with various laws, an employer must maintain records of each payroll run made for the firm. In response to this need, Ohio Scientific has developed the Personnel/Payroll System under OS-DMS.

The programs in this computerized system allow the operator to perform three basic personnel functions. The first is to compile and maintain an accurate employee data file, the second is to generate reports from this data, and the third is to run payroll.

The various primary options available in this system are as follows:

```
                PERSONNEL AND PAYROLL SYSTEM
-------------------------------------------------------------
(1) INSTRUCTIONS
(2) EMPLOYEE EARNINGS REPORT
(3) CHECK FILE REPORTS
(4) PAYROLL FOR ALL ACTIVE EMPLOYEES
(5) PAYROLL FOR INDIVIDUAL EMPLOYEES
(6) EDIT PAYROLL INFORMATION
(7) EDIT EMPLOYEE INFORMATION
(8) EDIT CHECK FILE INFORMATION
(9) SORT PAYROLL INFORMATION
(10) SORT EMPLOYEE INFORMATION
(99) EXIT
```

FIGURE 8. PERSONNEL/PAYROLL MENU

From the menu above, the various options available to the operator may be described as follows:

Option 1 presents to the operator a series of instructions as to how to run the system properly.

Options 2 through 5 are covered in detail following this explanation. Options 6 through 10 are file maintenance operations used to put new employees on the file, change the information in their records when needed and sort their records in various sequences as the occasions warrant. These runs, while not the producers of the end product, are used with regularity to maintain the various files. Options 4 and 5 perform the same operations with one primary difference: the payroll run for an individual does not update the computer files. When an out-of-cycle check is run for an employee, then the editing programs should later be used to enter the data into his records.

The sequence of operations for all payroll runs starts with the presentation to the operator of the run options. These are shown on the monitor screen and set the pattern for the entire run. Following this, the entries for the individual employees are made. The computer gets these through a dialogue with the operator as shown below.

```
EMPLOYEE NAME     ALCORN ALAN
EMPLOYEE NUMBER   0                    WAGE EMPLOYEE
    << NO COMMISSIONS >>
REGULAR HOURS WORKED ? 80
OVERTIME HOURS WORKED ? 2. 3
OTHER PAY ? 123. 11
```

FIGURE 9. EMPLOYEE HOURS ENTRY

With the information now known, the calculations of the individual's payroll figures are made and the following display of the results is made on the monitor screen:

```
EMPLOYEE NAME: ALCORN ALAN
EMPLOYEE NUMBER: 0                                  WAGE EMPLOYEE

REGULAR PAY:      400.00
OVERTIME PAY:      17.25
COMMISSION:         0.00
OTHER PAY:        123.22          GROSS PAY:        540.47
DEDUCTIONS
 FICA                                33.13
 FEDERAL INCOME TAX                  79.51
 STATE INCOME TAX                     6.01
 LOCAL TAX                            5.40
 INSURANCE                            7.50
                          GROSS DEDUCTIONS        131.55
                                   NET PAY        408.92

IS THIS CORRECT ? N
  (1) EARNINGS CHANGE (COMPLETE CHANGE)
  (2) DEDUCTION CHANGE
  (3) SKIP THIS PERSON
? 2
```

FIGURE 10. EMPLOYEE PAYROLL CALCULATIONS VERIFICATION

If the employee deduction change is desired, the following series of questions and answers takes place between the operator and the computer.

```
                    POSSIBLE DEDUCTIONS
1 FICA                    2 FEDERAL INCOME TAX
3 STATE INCOME TAX        4 LOCAL TAX
5 INSURANCE               6 OTHER DEDUCTIONS
                    SELECT DEDUCTIONS
FICA
  IS THIS DEDUCTION TO BE USED ?
FEDERAL INCOME TAX
  IS THIS DEDUCTION TO BE USED ?
STATE INCOME TAX
  IS THIS DEDUCTION TO BE USED ? N
LOCAL TAX
  IS THIS DEDUCTION TO BE USED ? N
INSURANCE
  IS THIS DEDUCTION TO BE USED ? N
OTHER DEDUCTIONS
  IS THIS DEDUCTION TO BE USED ? Y
ENTER THE AMOUNT OF THE EMPLOYEE DEDUCTION ? 50.00
```

FIGURE 11. EMPLOYEE DEDUCTION CHANGE

At the end of the run the totals for the entire payroll run are made. These are needed f several reasons, most importantly to insure that the amount of money being paid out known and can be deposited in the payroll bank account to cover the checks being issue

```
PAYROLL FOR 7/3/79          TODAYS DATE: 7/20/79          PAGE: 1
=============================================================
TOTAL NUMBER OF EMPLOYEES PAID 4
AVERAGE OVERTIME HOURS 1.86
TOTAL REGULAR PAY      1749.35
TOTAL OVERTIME PAY       51.74
TOTAL COMMISSIONS         0.00
TOTAL OTHER PAY           0.00
                    TOTAL GROSS PAY      1001.09
TOTAL FICA             110.40
TOTAL FEDERAL          228.58
TOTAL STATE             15.97
TOTAL LOCAL             18.02
TOTAL INSURANCE         37.50
TOTAL OTHER DEDUC        0.00
                    TOTAL DEDUCTIONS      410.47
                    TOTAL NET PAY        1390.62
-------------------------------------------------------------
```

FIGURE 12. PAYROLL SUMMARY

When all the entries have been made and the totals printed, the check stubs may be print for the employees. A variation of this run is to have the computer print the actual check but since check designs vary significantly from company to company, the actual che printing program is not included as a part of the system.

For filing purposes, the payroll register is printed and is then available for historical us

```
CHECK FILE REPORT FOR THE 7/3/79 PAYROLL    TODAY'S DATE: 7/29/79    PAGE
=============================================================
OHIO SCIENTIFIC    7/3/79                OHIO SCIENTIFIC    7/3/79
NAME: ALCORN ALAN                        NAME: BEACHLY JERRY

REGULAR HRS.      73.75                  REGULAR HRS.      80.00
OVERTIME HRS.      2.50                  OVERTIME HRS.       .32
 PAY RATE         5.00                   PAY RATE         3.75

REGULAR PAY:     368.75                  REGULAR PAY:     300.00
OVERTIME PAY:     18.75                  OVERTIME PAY:      1.80
COMMISSIONS:       0.00                  COMMISSIONS        0.00
OTHER PAY:         0.00                  OTHER PAY:         0.00
 GROSS PAY               387.50           GROSS PAY               301.80
-------------------------------------------------------------
DEDUCTIONS                               DEDUCTIONS
FICA:             23.75                  FICA:             18.50
FEDERAL:          48.33                  FEDERAL:          32.90
STATE:             2.95                  STATE:             2.05
LOCAL:             3.88                  LOCAL:             3.02
INSURANCE:         7.50                  INSURANCE:         0.00
MISC:              0.00                  MISC:              0.00
-------------------------------------------------------------
TOTAL DEDUCTIONS          86.41          TOTAL DEDUCTIONS          56.47
NET PAY                  301.09          NET PAY                  245.33
-------------------------------------------------------------
```

OHIO SCIENTIFIC 7/3/79
NAME: CALOWAY JOHN

REGULAR HRS.	80.00	
OVERTIME HRS.	0.00	
PAY RATE	733.83	
REGULAR PAY:	733.83	
OVERTIME PAY:	0.00	
COMMISSIONS:	0.00	
OTHER PAY:	0.00	
GROSS PAY		733.83
DEDUCTIONS		
FICA:	44.98	
FEDERAL:	87.90	
STATE:	8.15	
LOCAL:	7.34	
INSURANCE:	22.50	
MISC:	0.00	
TOTAL DEDUCTIONS		170.87
NET PAY		562.96

FIGURE 13. CHECK STUBS

ECK FILE REPORT FOR THE 7/3/79 PAYROLL TODAY'S DATE: 7/29/79 PAGE: 1

AME: ALCORN ALAN					
EG HRS: 73.75	REG PAY:	368.75	FICA:	23.75	
VT HRS: 2.5	OVT PAY:	18.75	FEDERAL:	48.33	
	COM PAY:	0	STATE:	2.95	
Y RATE: 5	OTHER PAY:	0	LOCAL:	3.88	
			INSURANCE:	7.5	
			OTHER:	0	
	GROSS PAY:	387.50	GROSS DEDUC:	86.41	NET PAY: 301.09
AME: BEACHLY JERRY					
EG HRS: 80	REG PAY:	300	FICA:	18.5	
VT HRS: .32	OVT PAY:	1.8	FEDERAL:	32.9	
	COM PAY:	0	STATE:	2.05	
Y RATE: 3.75	OTHER PAY:	0	LOCAL:	3.02	
			INSURANCE:	0	
			OTHER:	0	
	GROSS PAY:	301.80	GROSS DEDUC:	56.47	NET PAY: 245.33
AME:CALOWAY JOHN					
EG HRS: 80	REG PAY:	733.83	FICA:	44.98	
VT HRS: 0	OVT PAY:	0	FEDERAL:	87.9	
	COM PAY:	0	STATE:	8.15	
Y RATE: 733.83	OTHER PAY:	0	LOCAL:	7.34	
			INSURANCE:	22.5	
			OTHER:	0	
	GROSS PAY:	733.83	GROSS DEDUC:	170.87	NET PAY: 562.96
AME: SMITH SANDY					
EG HRS: 77.06	REG PAY:	346.77	FICA:	23.17	
VT HRS: 4.62	OVT PAY:	31.19	FEDERAL:	59.45	
	COM PAY:	0	STATE:	2.82	
Y RATE: 4.5	OTHER PAY:	0	LOCAL:	3.78	
			INSURANCE:	7.5	
			OTHER:	0	
	GROSS PAY:	377.96	GROSS DEDUC:	96.72	NET PAY: 281.24

GURE 14. PAYROLL REGISTER

nother format used for saving and filing the information is that used for the earnings and eductions summaries. If there are few types of earnings and deductions, these two may be ombined as a single report.

AYROLL SUMMARY FOR 7/3/79 TODAY'S DATE: 7/29/79 PAGE: 1

AME	REG HRS	OVT HRS	PAY RATE	REG PAY	OVT PAY	COM	OTHER
LCORN ALAN	73.75	2.5	5	368.75	18.75	0	0
EACHLY JERRY	80	.32	3.75	300	1.8	0	0
ALOWAY JOHN	80	0	733.83	733.83	0	0	0
MITH SANDY	77.06	4.62	4.5	346.77	31.19	0	0

AYROLL SUMMARY FOR 7/3/79 TODAY'S DATE: 7/29/79 PAGE: 2

AME	FICA	FEDERAL	STATE	LOCAL	INSURANCE	OTHER
LCORN ALAN	23.75	48.33	2.95	3.88	7.5	0
EACHLY JERRY	18.5	32.9	2.05	3.02	0	0
ALOWAY JOHN	44.98	87.9	8.15	7.34	22.5	0
MITH SANDY	23.17	59.45	2.82	3.78	7.5	0

IGURE 15. EARNINGS AND DEDUCTIONS SUMMARIES

When ends of reporting periods pass, various reports are required for numerous governmental agencies as well as the firm's own needs. To provide the data necessary for the production of these reports, the employee earnings record is often used. This report prints all payroll information about a given employee or about all employees for whatever payroll or group of payrolls is specified at the start of the program. It is formatted as follows:

NAME: JOHN DOE
ADDRESS: 1234 SOUTH LAKE STREET
KENT, OH 44240
PHONE: 898-1121

PERIOD ENDING	REG HRS	OVT HRS	TOTAL HRS	PAY RATE	REG PAY	OVT PAY
1/2/79	57.00	0.00	57.00	0.00	456.00	0.00
1/16/79	87.75	0.00	87.75	8.00	702.00	0.00
1/30/79	73.00	0.00	73.00	8.00	584.00	0.00
2/13/79	82.00	0.00	82.00	8.00	656.00	0.00
2/27/79	86.50	0.00	86.50	8.00	692.00	0.00
3/13/79	88.50	0.00	88.50	8.00	708.00	0.00
3/27/79	70.00	0.00	70.00	8.00	560.00	0.00
4/10/79	76.50	0.00	76.50	8.00	612.00	0.00
4/24/79	85.00	0.00	85.00	8.00	680.00	0.00
5/8/79	83.50	0.00	83.50	8.00	668.00	0.00
5/22/79	86.50	0.00	86.50	8.00	692.00	0.00
6/5/79	129.25	0.00	129.25	8.00	1034.00	0.00
6/19/79	80.00	0.00	80.00	8.40	672.00	0.00
7/3/79	86.00	0.00	86.00	8.40	722.40	0.00

TOTALS FOR THE PERIOD

REG. HRS.:	1171.50	REG. PAY:	9438.40
OVT HRS:	0.00	OVT. PAY:	0.00
		COM. PAY:	0.00
		OTHER PAY:	466.02
		GROSS PAY:	9904.42

EMPLOYEE EARNING RECORD
SOCIAL SECURITY NUMBER: 111-12-2222

DATE HIRED: 1/4/74
DATE TERMINATED:
OCCUPATION: ASSEMBLER
MARITAL STATUS: SINGLE
ALLOWANCES: 1

OTHER PAY	GROSS PAY	FICA W.T.	FED W.T.	STATE W.T.	LOCAL W.T.	OTHER W.T.	GROSS DEDUC	NET PAY
0.00	456.00	27.59	70.35	3.82	4.56	0.00	106.32	349.68
114.54	816.54	50.05	181.65	12.19	8.17	0.00	252.06	564.48
0.00	584.00	35.80	102.97	6.38	5.84	0.00	150.99	433.01
59.29	715.29	43.85	143.05	9.56	7.15	7.50	211.11	504.18
0.00	692.00	42.42	134.70	8.98	6.92	7.50	200.52	491.48
107.81	815.81	50.01	176.46	12.17	8.16	7.50	254.30	561.51
0.00	560.00	34.33	96.08	5.90	5.60	0.00	141.91	418.09
0.00	612.00	37.52	110.70	6.98	6.12	7.50	168.82	443.18
89.18	769.18	47.15	160.60	10.91	7.69	7.50	233.85	535.33
0.00	668.00	40.95	127.50	8.38	6.68	7.50	191.01	476.99
14.71	706.71	43.32	139.37	9.35	7.07	7.50	206.61	500.10
0.00	1034.00	63.38	255.70	12.90	10.34	7.50	349.82	684.18
80.49	752.49	46.13	154.93	10.49	7.52	7.50	226.57	525.92
0.00	722.40	44.28	144.70	9.74	7.22	7.50	213.44	508.96

FICA W.T.:	606.78
FEDERAL W.T.:	1998.76
STATE W.T.:	127.75
LOCAL W.T.:	99.04
OTHER W.T.:	75.00

GROSS DEDUCTIONS: 2907.33 NET PAY: 6997.09

FIGURE 16. EMPLOYEE EARNINGS RECORD

The next issue will conclude our series of articles on OS-DMS with a discussion of the new Educational Testing System, Quotation/Estimation System and the enhanced Inventory System including order entry, inventory explosion and forecasting.

OHIO SCIENTIFIC 1333 S. Chillicothe Road • Aurora, Ohio 44202 • (216) 562-3101

REVIEWS

(from page 9)

such as CASE statement, scalars, subranges and sets. Chapter 6 introduces arrays and records; chapter 7 presents files. Titled "Dynamic Data Structures," chapter 8 covers pointers and linked lists in one of the most lucid treatments I've ever seen. Chapter 9 finishes up the language with the GOTO statement and some discussion of dynamic memory allocation.

As soon as you have some grasp of the language, you can read chapter 10, which covers program design. This is handy for the beginner who may not have any idea at all how to go about developing, testing and debugging a program. For the non-programmer who wants to learn how, this chapter alone is worth the price of the book. Appendix D, "Program Standards," should be read along with this chapter for a good understanding of what makes a program a good or bad piece of work.

A section of the book simply titled "Further Reading," with no chapter or appendix number, is a boon to somebody like me who knew absolutely nothing about PASCAL before he picked up this book. This section contains a good PASCAL bibliography along with information on the PASCAL Users' Group. Used in conjunction with your local college library, this section will easily take care of your spare time for about a month.

Appendices A and B cover, in summary, the vocabulary and syntax of PASCAL. The syntax appendix (B) contains all those marvelous syntax diagrams that are spread throughout the text. I've copied mine and have them pinned to the wall in front of my desk for quick reference while I'm still learning PASCAL. Appendix C, the last of the appendices, can be safely ignored unless you happen to be using the CDC implementation of PASCAL. It contains a discussion of the peculiarities of that implementation and is rather interesting, but probably not much use to most readers of the book.

After raving about the book for eight paragraphs I have to register my one complaint. Virtually all the examples in the early part of the book are mathematical. So are the exercises at the end of the chapters. Despite being a computer-science major, I feel that an introductory book like this should make no assumptions about the reader's background. A programming text *not* directed toward embryonic engineers will be welcome. Maybe someday I'll write my own.

William L. Colsher
Lisle IL

Computer Power for the Small Business
Charles J. Sippl, Fred Dahl
Prentice-Hall, Inc.
Englewood Cliffs NJ
1979, $15.95

This book is full of words. Virtually all of them are arranged into grammatically correct sentences. Virtually none of these sentences can convey any useful information to the intended reader, the small businessman.

If we may begin with a summary, this is the worst "technical" book I have ever seen. The writers don't understand the technology they attempt to report on (in much too much detail); they don't understand the requirements for business computing; and there is no way that their intended reader, the small-business "operator" (their term), could get past the flood of meaningless (to him) technical detail to what little good advice the book contains.

To support these contentions, let's look at the authors' reporting of computer technology. "A 12-inch rectangular CRT *usually* displays 32 characters in 8 lines." (Emphasis mine.) Now, they are supposed to be talking about business computers, including micros, but that description fits the crudest of TV games at best. "Usually?"

"The *standard floppy* is 9 inches square." (The italics are theirs this time, saving me the trouble!) If floppies were nine inches on a side they would loose one of their finest attributes: the ability to be shoved into a "standard" file folder in a "standard" file cabinet. It makes me wonder if the authors ever saw a floppy, or a ruler.

While on the subject of floppies, the writers warn the reader to be on the lookout for "preformatted" floppies, which they insist means "hard-sectored." (Just the opposite is the case, of course.) Their warning is ridiculous to begin with, as they claim that preformatting will force the user to adapt his file size to that required by the preformatting. File sizes are, of course, under control of the operating system, and are invariably quite flexible.

I am not trying to nitpick. I want to make it plain that these are only a couple of examples of the abysmal lack of technical expertise displayed by the authors— which makes their poor attempts to explain technology to the naive reader a real imposition on his valuable time and a waste of his $15.95. And they don't know when to quit, roaming far afield from the subject of business computers to discuss everything from microphones to robots, in great detail and with gross inaccuracy.

I never knew that those funny little bar codes on packages in the grocery store are what is known as the "Universal Produce Code." Just try to stick one on a head of lettuce!

Moving on to the subject of the requirements for a business computer, the picture caption on page 77 opens with the statement that "RCA's Cosmac VIP is certainly applicable to business uses . . . ." The photo shows a gentlemen laboriously keying in what appears to be a multiple-page program through the VIP's tiny keypad. Now, the VIP is a fine little machine as far as it goes, and it can be used as an introduction to micros; as the basis on which to build a hobby computer; or as a toy. But a business computer it certainly is not!

The writers just don't seem to know what computing or data storage a business requires. Hence their statement that "a year's worth of records can very likely be contained on—at most—two floppies." Did they ever interview anyone using a computer in business? What research led to this astounding conclusion? A year's worth of records for an entire small business on two floppies? It would seem to be a wonderful dream. Maybe it was.

Similarly detached from reality is the statement, in the context of discussing the smallest ($2000) systems, that software costs "can be as much as half the cost of the hardware." This statement appears to be based on a single example of a system used to control an air-conditioning system—not a business computer as such. It is offered to a reader contemplating the purchase of a business computer as advice on what to expect his software to cost. At best it is a gross underestimate.

It is in the field of business software that the prospective computer user needs the most advice, and the most accurate advice. Unfortunately, this book contains almost none. The chapter on software lists computer *languages* without explaining their features or intended uses. It ignores the subject of application programs, which is of paramount importance to the business user.

In summary, the writers have tried to cover an incredibly bewildering array of technical topics in a book intended to be a guide to selecting a business computer. And they do cover these topics—with an impenetrable rubble of unnecessary definitions and technical inaccuracies, preventing the businessman from finding what little good advice the book does contain. But the silly advice overwhelms the good . . . for instance, the suggestion that the businessman should read such publications as "Digital Design," which is exactly what its name implies, a periodical for digital-design engineers.

One accurate statement that the book does contain is that the book itself (ignoring pictures) "could easily be contained on a single floppy diskette." Perhaps the book *should* be on a floppy. It would make it easier to return it to the seller. And it should be returned, on the grounds that it is not at all what it is advertised to be: "a complete and practical guide" to selecting and using a business computer.

But save the return postage. Don't buy this book.

One final, ironic quote from Sippl and Dahl: "Something is missing in the business computer market: knowledgeability." They provide a prime example of the truth of that statement!

Ken Barbier
Borrego Springs CA

Logic—Home Study Course
Milton Howard Aronson
Measurement & Control
Pittsburgh PA
32 pages, Paperback, $3

If you are in any way mixed up with computers or logic devices, you probably should have this course. The course is actually a "home study" article reprinted from the September-October 1970 issue of the trade magazine *Medical Electronics and Data.* It begins by explaining what logic is and what it does. Assuming no particular background in either

(see REVIEWS, page 108)

ESFOA Newsletter

Secretary, Fred Waters

STRINGY FLOPPY FOR SWTP

This month the newsletter of the Exatron Stringy Floppy Owners Association (ESFOA for short) addresses the SWTP version. in keeping with the theme of this issue.

If you haven't taken a close look at the front cover, do it now. You can see that the Exatron Stringy Floppy is capable of earning a living at the office as well as in the more traditional role as "Keeper of the Games". This exciting new subsystem consists of a controller and up to two transports. Combining the economy of tape with the speed and reliability of the disk, Exatron gives you another choice.

The CONTROLLER is a complete system on a board, comprising a synchronous serial interface, a data encoder, a clock recovery circuit, and the necessary latches for peripheral control. Also on board are EPROMs containing the software drivers. There are four sockets for 2708s; three are used for the basic system software. The EPROMs occupy memory addresses $C000–$CFFF in the standard configuration. However the controller is addressable to any 4K block in memory, and you can order custom EPROMs in any block to fit your system.

UTILITIES firmware consists of the low-level drivers and an I/O package designed to reduce monitor dependency. No calls to a monitor are used: I/O is accomplished by direct manipulation of a serial interface for the console, and a parallel interface for the printer. The console I/O is done through a vector which may be changed after initialization. The board cold-start routine sets the vector at Port 2 if there is nothing at Port 1. The printer is set at Port 7. The utilities program has many useful subroutines which can be used by other programs.

SYSTEMS SUBROUTINES are also contained in the firmware. The command TBAS will load and execute TSC 6800 BASIC, the fastest BASIC running on an 8-bit microcomputer. The command MBAS will load and execute Microsoft BASIC, which has data file capability, as well as compatibility with Microsoft BASICs on other machines. EDIT loads the TSC Text Editor, an excellent line- and content-oriented editor with many local and global commands. ASMB runs the TSC Assembler, a fast mnemonic assembler with object code generation into memory or into a Stringy Floppy.

Other system commands include HSAVE, HLOAD, NEWTAPE, and ASN. For a system having one drive, no drive number is specified in the command. If you have a two-drive system, you specify the drive number if the command addresses the drive not designated as the system drive. This designation can be changed using the ASN command. NEWTAPE is the routine for certifying new wafers. A fourth EPROM socket with room for another 1K of memory is available for system expansion.

The TRANSPORT uses state-of-the-art digital and linear techniques. It reads and writes 14,400 bits per second, or optionally 7200 bits per second, with a typical error rate of 1 in 100,000,000 bits. The transport has an average life of over 3500 hours. It has controls for motor on, fast forward, tape mark, wafer present, write protect, and write enable. Power is supplied by the controller.

The WAFER is a high-impact plastic case, about the size of a business card, and 3/16 of an inch thick, enclosing an endless loop of tape. The tape is wound somewhat like that in the familiar 8-track cartridge. Wafers are available with various lengths of tape: 5, 10, 20, and 50 feet. The average life of the wafer is 2500 passes.

PRICES

Cntrlr + Trnsprt	$250.00
Trnsprt only	$124.50
TSC 6800 BASIC	$39.95
Microsoft BASIC	$43.00
TSC Text Editor	$31.00
TSC Assembler	$33.00
Advanced Programmer's Guide	$20.00
Ctrlr, 2 Trspts, Micrsft, & TSC software	$499.95

JIM MAYNARD WORKING ON SOFTWARE FOR THE SS-50 ESF

NOW HEAR THIS! Membership in the ESFOA is automatic when you purchase an Exatron Stringy Floppy. One of the purposes of this Newsletter is to tell you about some of the interesting and useful projects which have been completed by ESFOA members. Jim Maynard, an expert programmer and hardware designer, has spent many many hours during the past year developing the SS-50 version of the ESF. Jim accomplished this monumental task singlehandedly and deserves a big round of applause from all SS-50 computer owners. Jim has some other exciting projects nearing completion.
Many other major software and hardware projects are in the works and will be reported on here in coming issues.

ESF FOR THE TRS-80

Have all of you TRS-80 owners out there seen the first two ESFOA Newsletters? In the August and September issues of Microcomputing? If you didn't, borrow a friend's, or go see your dealer, and read them–they describe the TRS-80 ESF. It's delivered assembled and tested at the factory, and within seconds of connecting it up you are ready to go. If you have any questions about it, or need more information, or want to order one, use our toll-free number. See the ordering information in the box below.

The ESF for the TRS-80 has met with great success. It fills an obvious need, and thousands of the TRS-80 owners were, in effect, looking for the ESF before they even knew of its existence. Disks are still expensive, and are needed when you have to store millions in bytes, but they are not what most personal computer owners are looking for. Audio cassette recorders and playback machines, along with audio-style cassettes, have their shortcomings, and YOU don't need to be told about that. [You should see some of our mail! Extracts from some of the letters, with horror stories and condemnation of cassettes and recorders are actually unprintable!] So ESFs are being shown by their owners with pride to their local TRS-80 Users Groups, and the word has spread.

You don't find many subsystems of this quality with an unconditional 30-day money-back guarantee. And there are none, we believe, with a one-year full warranty. To quote from the Users Manual, the specific purpose of the 30-day money-back guarantee is to ensure that all ESF owners are satisfied ESF owners.

HOW TO ORDER

All versions of the Exatron Stringy Floppy are covered by a 30-day moneyback guarantee and a one-year full warranty. The ESF is delivered from the factory assembled and tested. Prices for the SWTP version are given complete in the text above. The TRS-80 version is $249.50, which includes Microsoft Level III BASIC. Instant Software on ESF wafers are available at the list price. BUS-EX, the TRS-80 bus extension, is $15 for the 2-for-1 model, and $5 for each added connector. Wafers at 5, 10, 20 and 50 feet are $2.50 each and $20 for 10. Shipping and handling is $3.00 per order. CA residents add tax. Order fast by calling our toll-free number below and giving your Visa or MasterCharge number.

User's Manual and information package is available at no charge. Shipping and handling is $3.00.

If you have any questions about the product, about Exatron, or ESFOA, please call the Hot Line. Address letters to ESFOA, 3557 Ryder St., Santa Clara, CA 95051.

Stringy Floppy is a trademark of Exatron Corporation ✓E48

HOT LINE **800-538-8559**

WITHIN CALIFORNIA **(408) 737-7111**

**Instant Software on ESF wafers are available at the regular HST price*

PET-POURRI

Len Lindsay

Good news at last. Things are looking up at Commodore, and good news seems to be on the horizon. In my last PET-pourri I was a bit hard on Commodore. This was due to a lot of negative comments from other users as well as lack of information from Commodore. Commodore is now trying to remedy all that. Many new people are being hired. Training programs for PET dealers have now been started. With a good training course, PET dealers will now be able to help PET users locally with specific information and assistance. Final versions of the PET printer and dual-floppy manuals are now being polished and should be available soon. I've seen preliminary copies of these, and they are a vast improvement over the preliminary manuals.

Commodore has products that I believe are far superior to the others in the field. The problem is learning how to unleash the hidden power in the PET, printer and floppy. Full potential of the PET is now beginning to be realized. Both the PET printer and floppy include microprocessors as controllers. A complete PET system (computer, printer, floppy) can handle very sophisticated uses. In the future we should see some comprehensive, serious software available.

Some of this software is already being finalized. Commodore has just signed a contract with the author of a fantastic word processor. I saw a preliminary version and was amazed. It was easy to use and had features I thought were only available on dedicated word-processing systems. I should have a copy of it within the week, so next month I should be able to relate more about it.

Commodore Disk

Commodore's 2040 dual-disk unit holds a great deal of potential. The reliability problems with the earliest units have apparently been remedied, and the disk operating system has been improved. The trick is in learning to use the system. The preliminary manual that comes with the disks is not too helpful. The final manual is being completed now and should help a lot. The 2040 disk utilizes the PET's IEEE port and comes set up as device #8. The manual shows how to change this if you need to.

Presently, the 2040 disk works only with the new PETs. Soon Commodore will begin shipping new ROMs for the old PETs; then the 2040 disk will work with the upgraded old PETs. You can only write on the front side of each diskette, which holds about 170K, but you can use the reverse side of the diskette if you carefully cut out a notch on the left side similar to the notch on the right. Beware, however, that even the smallest particle allowed inside may ruin the diskette.

A significant improvement has been made in the Commodore disk operating system. DOS 3.1 is now included with every system. It takes all the drudgery out of operating the floppy. Take errors as an example. In the program mode, your program can have a subroutine to check for errors. The final manual lists routines that do this. In the direct mode, you only need to hit one key, > (greater than), and the error message is displayed on the PET screen.

The new DOS also eliminates the need to type OPEN and PRINT# commands for accessing the disks in direct mode. Now you simply type > followed by whatever you normally would have had to enclose in quotes in a PRINT# statement. Loading a program from disk is now easy. Just type / (slash) followed by the program name. DOS takes over and does all the work for you, including deciding which disk has the program. If you begin with the character ↑ (up arrow) instead of / the program automatically will run once loaded. And remember, the PET can use pattern matching for finding the program. The program Backgammon 3.4 can be loaded and run with the following direct command: ↑B*.

The asterisk tells the PET to load the first program that begins with the characters preceding the asterisk. In this case, PET would load the first program beginning with B. If another program, such as Baseball 2.5, preceded Backgammon 3.4 on the disk you could use: ↑BAC*. Once loaded into memory the program will automatically RUN.

Which Disk for Your PET?

Do you own several PETs, or do you also have an Apple and a TRS-80? If you answered no, then the Nestar dual-drive full-size floppy disk system is not for you. The Nestar Cluster One is ideal for schools. Up to 15 PETs, TRS-80s and Apples can hook up to one Cluster One. Contact Nestar (810 Garland Dr., Palo Alto CA 94303) for the name, address and phone number of your nearest Nestar dealers.

Computhink Disk and News

Most readers probably only have one PET. Two available disk systems, which I have seen work, are worth your consideration. One, of course, is the Commodore disk. The other is from Computhink (3260 Alpine Rd., Menlo Park CA 94025). Computhink's disk system has been out for over a year already. Readers in the East and Midwest should contact New England Electronics (NEECO), 679 Highland Ave., Needham MA 02194, the Computhink distributor for that section of the country. NEECO now has a new disk operating system as well as a dual-density option. Both sides of each diskette may be written on, unless the disk protect notch is covered, then the diskette cannot be written on. With dual density, each diskette can hold about 200K. Presently, the Computhink disk will only work with the old PETs; soon, it should be available for the new PETs as well.

The Computhink disk system is designed to be plugged into Computhink's Expandamem. It utilizes the PET memory expansion port rather than the IEEE port. The system comes with a diskette containing utilities on one side and helpful examples of disk programming on the other. The utilities include: Monitor, Datafile, Diskcopy (to copy a complete diskette from one drive to the other), Random-Format (sets up a direct-access file), DensityCopy (allows a dual-density drive to read a single-density diskette and create a dual-density copy of it) and two memory tests, PageTest and BlockTest. The examples on the other side are very helpful in showing how to use the commands explained in the manual.

Currently, Computhink is shipping dual-density, dual minifloppy drive systems for the PET. By early fall 1979, they should be shipping this configuration with an added bonus: You can read both sides of each diskette without having to take it out and flip it over, thus having four sides of dual-density diskettes on line—or 800K on line.

Computhink offers more than hardware. They also have a PL/M compiler available for their disk system. A BASIC and FORTRAN compiler will be available soon. Also available now is a comprehensive data base. Contact Computhink for details on these.

Computhink Disk Notes

In the short time that I've had the Computhink dual-density disk system I've developed a Menu program, written for the new DOS, that will enable you to load and run any program on either drive by hitting only one key. The program first puts a directory on the screen. Then an index-key character is printed in front of each entry. The program checks which key is hit and goes to the correct location on the screen. It PEEKs at what name is there and assigns it to F$. It then loads and runs F$ from drive D. The program keeps track of which drive the directory came from and allows you to hit one key to see the directory from the other drive. If you modify each of your programs to end by loading Menu, you can have a continual chain taking almost all the work out of running programs.

I'm not a professional programmer. My program is a bit long and can be written in a more condensed form, but the program now includes many remarks and will be helpful in developing your own version.

Disk Software

Business Software is available for the Computhink disk from

NOW... Exclusively Yours from ACS

TRS-80®-DOS 3.0

Disk Operating System by the Original Author

☐ DOS 3.0
EFFECTIVE SEPTEMBER 1
ONLY **$49.95**

Check these DOS 3.0 features . . . that go far beyond 2.1

NO KEY BOUNCE • UPWARDS COMPATABLE FROM 2.1 • CHECK EOF
SEEK • RE-READ • RE-WRITE • LOC • VARABLE LENGTH RECORDS
SKIP • DISK LOGGING OF MESSAGES • BOOT • CHAIN • WRITE EOF
PURGE • SET • RESET • ROUTE • RUN & LOAD FOR 1 DRIVE SYS.
XFER • FORMAT W/O ERASE • DIR FROM BASIC • PATCH • LINK
USER DEFINED KEYS • KEY AUTO REPEAT • UPPER AND LOWER
CASE DRIVER • SHIFT LOCK • RS-232-C DRIVERS
•PAUSE •USER DEVICE CONTROL BLOCKS
•MULTI PROTOCOL COMMUNICATIONS

HARDWARE FOR TRS-80®

☐ **Pertec Disk Drives FD-200** ..... **$375.00 ea.**
These are 40-track Drives that are completely compatible with the TRS-80® and Radio Shack Drives. 3.0 DOS included. Will allow Turning Diskette over and Write on other side.

☐ **4-Drive Cable for Pertec Drives** **$35.00**

☐ **DECwriter III, 132 Character** **$2500.00**
110 to 9600 band EIA tractor feed keyboard printer. This is truly the nicest printer available. (30 day delivery)

☐ **701 Centronics TRACTOR FEED Bidirectional Printer** **$1500.00**
2½ times as fast as the Radio Shack 779 Printer, has full size 132 Char. Carriage Bell tone. Complete with Cable plug in and use. Shipped Freight COD.

☐ **200 ns 16K Dynamic Memory Clips for Keyboard or Expansion Interface, Lifetime Guarantee, complete** **$110.00**
Lifetime Guarantee. Complete with Instructions and Jumper Blocks.

☐ **10 Key Numerical Keypad Kit** **$79.95**

☐ **TRS-80® Level II – 16k** **$750.00**
- ☐ **Expansion Interface** **$275.00**
- ☐ **RS-232-C Interface** ..... **$ 89.00**

ORDER NOW AND SAVE

Just list the items you want and mail this convenient coupon.

SOFTWARE BY ACS

☐ **Monitor No. 3** **$29.95**
Complete Machine Language Monitor for TRS-80® features: Find, EDIT, Relocate, Symbolic Dump to Tape, etc.

☐ **Monitor No. 4** **$49.95**
All of the commands that reside in Monitor No. 3, plus: RS-232 I/O, Disk Program I/O, Symbolic Dump to Disk for Loading into Disk Editor/ASM., Track & Sec I/O for modification.

☐ **PCLEND** **$15.95**
Will Patch ASCII files of Basic Programs or text or DATA FILES so that they may be loaded into the Disk Version of the Electric Pencil for Editing purposes comes on Cassette that will automatically create a Disk file of PCLEND.

☐ **MAKE TAPE AND MAKE DISK for Cassette Dealers** **$69.95**
These are two programs that will allow you to take any type of Program from Disk and store it on tape for mailing purposes. When the user receives the program in the mail on cassette, it is loaded into the computer which will automatically make a Disk file of the program.

☐ **CP/M & C BASIC for the TRS-80®**
CP/M Includes: MOVCPM, STAT, PIP, Dump, DDT, ASM (8080), ED, plus 6 user manuals.
CP/M **$150.00**
C Basic-2 Includes: XREF2, CBAS2, and manuals.
C BASIC 2 ..... **$99.95**

☐ **G2 LEVEL III BASIC for TRS-80®** ..... **Special $39.95**

☐ **TELCOM – Telecommunications for the TRS-80®** **$29.95**
Telecommunications for the TRS-80® allows one TRS-80® to communicate with another through the RS-232-C over the phone line.

® TRS-80 is a registered trademark of Radio Shack, a division of Tandy Corp.

Orders received by 6:00 p.m. shipped next day on Master Charge, Visa, Certified Check or Money Order. Personal Checks require 14 days to clear. No C.O.D. Collect calls not accepted. All Hardware warranted for 90 days except Radio Shack equipment which is warranted through Radio Shack. Software guaranteed for replacement only. Prices subject to change without notice.

AUTOMATED COMPUTER SOFTWARE SERVICE A75
(615) 244-2798
Division of
Computer World INC.
625 Main Street • Nashville, TN 37206

Send Check or Money Order payable to –

SOFTWARE • P.O. Box 60097 • Nashville, TN 37206

Quan.	Description	Unit Price	Total
	HANDLING CHARGE $1.50 TENN. RES. ADD 6% SALES TAX **TOTAL**		

☐ Check
☐ Money Order
☐ MasterCharge
☐ Visa
Card No. _____
Exp. Date _____

Name _____

Address _____

City _____ State _____ Zip _____

```
  0 REM  THIS IS MENU FOR DISK
  1 REM  IT HANDLES UP TO 39 FILES
  2 REM  IT HANDLES ALL CHARACTERS IN
  3 REM  NAMES EVEN SIMPLE REVERSE
  4 REM  FIELD IF THE REVERSE FIELD
  5 REM  IS TURNED OFF WITHIN THE NAME
  6 REM
  7 REM
 10 INPUT "[CLR][ 2 DOWN][ 2 RIGHT]WHICH DRIVE#  1[ 3 LEFT]";D
 20 POKE 525,1:POKE 527,99
 30 $D,D
100 PRINT "[HOME]&&&HIT THE [RVS]KEY[OFF] OF YOUR PROGRAM CHOICE&&&"
110 PRINT "[UP]&&&HIT [RVS]*[OFF][LEFT][DOWN]"[UP] TO STOP - HIT [RVS]>[OFF][DOWN][LEFT]"[UP] FOR DRI
190 M=32888:REM FIRST INDEX LOCATION
195 F=0:REM INITIALIZE END FLAG
197 IK=0:REM INDEX KEY INITIALIZE
200 FOR L=0 TO 19
205 IF F=1 THEN GOSUB 9200:GOTO 290:REM END HAS BEEN FOUND ALREADY
215 IF PEEK(M+L*40+2)=134 THEN F=1:GOTO 290:REM THIS IS THE FREE TRACK LINE
220 POKE M+L*40,L+129:REM THIS IS THE INDEX LETTER
230 POKE M+L*40+1,31:REM THIS IS A "_"
290 NEXT L
294 L=L+1:GOSUB 9200
300 IF F=1 THEN 1000:REM SKIP SECOND COLUMN - ALREADY DONE
305 M=32907:IK=0:REM SECOND COLUMN
310 FOR L=0 TO 5:REM FOR UVWXYZ
320 IF F=1 THEN GOSUB 9200:GOTO 390
330 IF PEEK(M+L*40+2)=134 THEN F=1:GOTO 390
340 POKE M+L*40,L+149:REM THIS IS THE INDEX LETTER
350 POKE M+L*40+1,31:REM THIS IS A "_"
390 NEXT L
400 IF F=1 THEN 1000:REM END FOUND ALREADY
410 FOR L=6 TO 15:REM FOR 0123456789
420 IF F=1 THEN GOSUB 9200:GOTO 490
430 IF PEEK(M+L*40+2)=134 THEN F=1:GOTO 490:REM END FOUND
440 POKE M+L*40,L+170:REM THIS IS THE INDEX NUMBER
450 POKE M+L*40+1,31:REM THIS IS A "_"
490 NEXT L
500 IF F=1 THEN 1000:REM END FOUND ALREADY
510 FOR L=16 TO 18:REM FOR @$%
520 IF F=1 THEN GOSUB 9200:GOTO 590
530 IF PEEK(M+L*40+2)=134 THEN F=1:GOTO 590:REM END FOUND
540 POKE M+L*40,L+147:REM THIS IS THE INDEX CHARACTER
550 POKE M+L*40+1,31:REM THIS IS A "_"
590 NEXT L
1000 GET AA$:IF AA$="" THEN 1000:REM GET CHOICE
1010 IF AA$="*" THEN 9990
1020 IF AA$=">" THEN D=2-D+1:GOTO 20:REM CHANGE DRIVES
1100 A=ASC(AA$)
1200 IF A>64 AND A<85 THEN M=32890:L=A-65:GOTO 1300:REM SKIP 2ND COLUMN
1205 M=32909:REM FOR SECOND COLUMN
1210 IF A>=85 AND A<=90 THEN L=A-65-20:GOTO 1300
1220 IF A>=48 AND A<=57 THEN L=A-42:GOTO 1300:REM FOR0123456789
1230 IF A>=35 AND A<=37 THEN L=A-19:GOTO 1300
1240 PRINT "[HOME][ 3 RIGHT][RVS]PLEASE CHOOSE ONLY AN INDEX LISTED[OFF]"
1250 GOTO 1000
1299 END
1300 F$="":RV=0:REM INITIALIZATION
1305 PRINT "[HOME][ 23 DOWN][ 3 RIGHT]>>>>>>>>>>>>>>[RVS]HERE IT COMES... [OFF]"
1310 FOR K=0 TO 15
1320 FL=PEEK(M+L*40+K):REM VALUE OF CHARACTER
1322 IF FL>128 AND RV=0 THEN GOSUB 9000
1324 IF FL<129 AND RV=1 THEN GOSUB 9500
1326 IF FL>128 AND RV=2 THEN GOTO 9600
1328 IF FL>128 AND RV=1 THEN FL=FL-128
1330 GOSUB 8000:REM CONVERT TO STRING CHARACTER
1390 NEXT K
2000 IF RV THEN GOSUB 9700
2010 $X,D,F$
8000 REM CONVERT PEEK TO CHR$
8002 REM FL IS PEEK VALUE
8004 REM FT IS CONVERTED FOR CHR$
8006 REM F$ IS THE STRING CHARACTER
8010 IF FL>=0 AND FL<=31 THEN FT=FL+64
8020 IF FL=>32 AND FL<=63 THEN FT=FL
8030 IF FL>=64 AND FL<=95 THEN FT=FL+128
8040 IF FL>=96 AND FL<=127 THEN FT=FL+64
8100 IF FL<128 THEN F1$=CHR$(FT):GOTO 8800
8800 REM F1$ IS CHR$ FOR THIS CHARACTER POKE
8810 F$=F$+F1$
8999 RETURN
9000 RV=1:REM REVERSE FOUND
9010 F$=F$+"[RVS]"
9020 FL=FL-128
9099 RETURN
9200 IF IK=1 THEN 9299
9210 IK=1
9215 POKE M+L*40-40,30:REM  ^
9220 POKE M+L*40,137:REM    I
9225 POKE M+L*40+1,142:REM  N
9230 POKE M+L*40+2,132:REM  D
9235 POKE M+L*40+3,133:REM  E
9240 POKE M+L*40+4,152:REM  X
9245 POKE M+L*40+5,160:REM
9250 POKE M+L*40+6,139:REM  K
9255 POKE M+L*40+7,133:REM  E
9260 POKE M+L*40+8,153:REM  Y
9299 RETURN
9500 RV=2:REM REVERSE FOUND AND TURNED OFF
9510 F$=F$+"[OFF]"
9599 RETURN
9600 PRINT "[HOME][ 2 RIGHT][RVS]YOUR FILE NAME IS TOO CONFUSING FOR ME[OFF]"
9610 GOTO 1000
9700 F$=LEFT$(F$,16)
9799 RETURN
9990 PRINT "[HOME][ 21 DOWN]"
9992 POKE 525,9
9994 POKE 527,145:REM UP
9996 FOR Z=528 TO 533:POKE Z,32:NEXT Z:REM SPACE
9998 POKE 534,145:POKE 535,141:REM SHIFT RETURN
9999 END
```

Menu program (contact author—not c/o Microcomputing, *please—for missing part of line 110).*

both Dr. Daley (425 Grove Ave., Berrien Springs MI 49103) and Sawyer Software (201 Worley Rd., Dexter MO 63841). Business Software for the Commodore disk is available from both NEECO and CMS Software (5115 Menefee Dr., Dallas TX 75227). Next time I hope to have more information on these.

It appears that both Dr. Daley and Programma International (3400 Wilshire Blvd., Los Angeles CA 90010) are offering programs on Computhink diskettes as well as on cassettes. (A software announcement for Programma in *Micro* mentioned this, and Dr. Daley notes this option in his brochures.) TIS (PO Box 921, Los Alamos NM 87544) has just announced that their text-editing program, MEDIT, is now available for use with Commodore disk and printer. It is priced at $12.95 and includes an over-40-page manual. It will be interesting to see which disk system will be supported by each of the many PET companies.

New PET to Printer Update

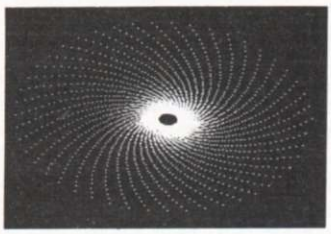
The new PET is a big improvement over the old PET. I haven't yet used a PET printer, but I'm told that it can print uppercase and lowercase letters as they appear on the screen. I don't know why Commodore didn't do this in their 2040-floppy manual. The good news on the manual is that it is only temporary; a final improved version is now being completed. There's a problem in using a printer such as the Teletype 43 via RS-232. Apparently the old PET did not use true ASCII code, and the RS-232 interfaces corrected for this. The new PET sends true ASCII code, which is a good improvement, but either the interface made for the old PET will have to be modified or a program written to convert uppercase to lowercase. If anyone has the answer, please contact me.

Upgrades for Old PET

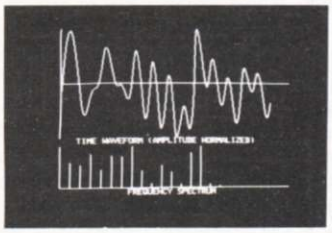
Old-PET owners now may buy a set of new ROMs for their old PET. There are two types of ROMs. Both are packaged in a set of seven ROMs and are priced at $89.95 per set from Commodore (3330 Scott Rd., Santa Clara CA 95050). To upgrade your old PET, simply remove the old ROM set and replace it with the new

OUR VISIBLE MEMORY ADDS FUNCTION AND FUN TO YOUR 6502

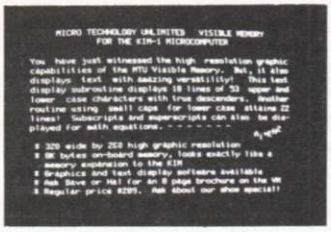
This DOT MATRIX display board doubles as an 8K MEMORY. You can use it as a display, memory expansion or both with graphic and text display software available. You get resolution graphics with no wait states, no snow, and no processor overhead. K-1008 ASSEMBLED and TESTED $240.00, BARE BOARD $40.00 K-1008-1 GRAPHIC/TEXT UTILITY SOFTWARE LISTING $20.00.



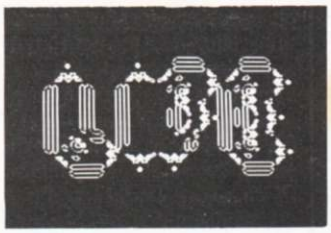
As a dot matrix formatted 200 high by 320 wide, it allows high resolution patterns to be displayed and evaluated. It enhances system performance for data acquisition displays, math equation plotting, etc.



For charts and graphs, it is valuable in the educational, research and business fields. Shown to the right is a single period of a complex frequency waveform and its frequency spectrum chart.



For text display, dot matrix is the ultimate. Any character set the user desires is possible. Even subscript and superscript in the same column, and—mixed text and graphics. The improved man-machine interface for an unskilled operator increases productivity.



Graphics allows you to program games with much more accurate representation of your game because YOU DETERMINE THE SHAPES on the display, not a ROM character generator. Our LIFE program allows tremendous creativity in creating complex colonies for observation of generation growth and decay.

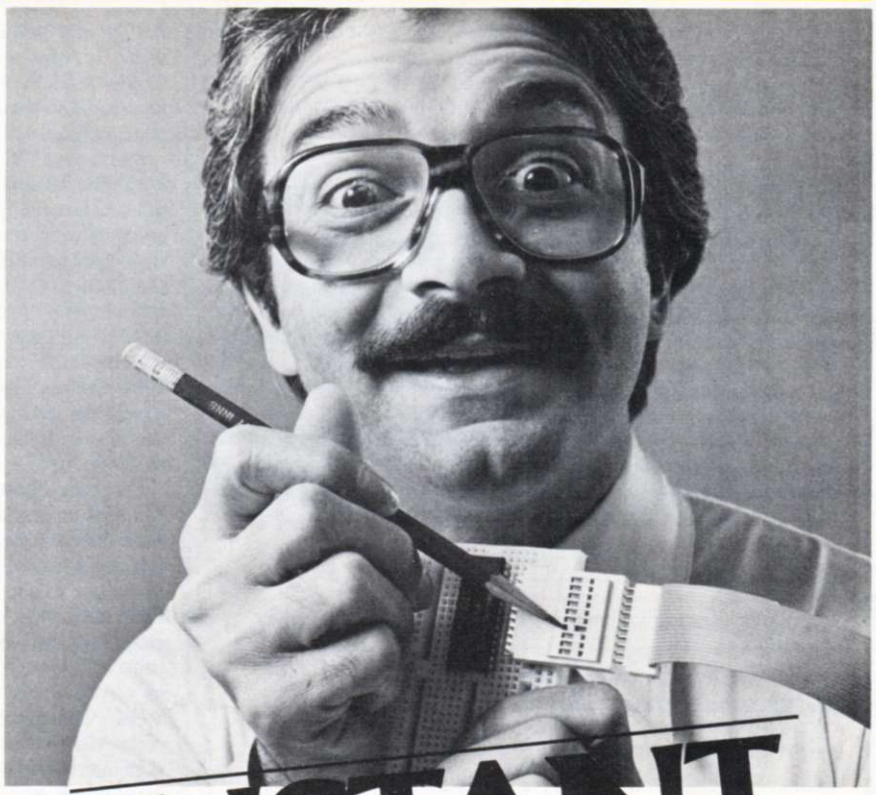
Developed by a leading group of New England engineers for interface with the three 6502 KIM BUS systems (KIM, VIM, AIM). The K-1008 requires only the processor card and its power supply to function as a memory. Add a monitor and you have graphics and text display available. Micro Technology also makes K-1000 Power Supplies, K-1005 Card Files, K-1002 8 Bit Music Systems (with advanced software), K-1020 Regulated Wire Wrap card, K-1012 IO/COMM/PROM/2708 Programmer card, K-1016 16K Memory, all sold assembled and tested or as bare boards. Write for technical information, prices and terms.

Overseas orders—include $3.00 for shipping.

Micro Technology Unlimited

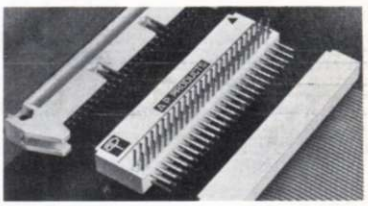
✓ M44

P.O. Box 4596 Manchester NH 03108



INSTANT.

The way you check line-by-line with an A P Intra-Switch or Intra-Connector.



You plug your Intra-Switch in-line with standard socket connectors, and instantly you've got a separate, independent on-off switch for each and every line in your flat ribbon cable. To switch, you nudge with a pencil point. It's that quick.

Imagine how much time and trouble Intra-Switch will save you in your diagnostic and quality testing, your programming and selective line inhibiting.

Or, plug in your Intra-Connector (see box) the same way, and you have an extra set of male contacts at right angles. Instant line-by-line probeability—and an easy way to tap your system and daisy chain it into new areas.

Both Intra-Connectors and Intra-Switches come in 20, 26, 34, 40 and 50-contact models.

Where? At your nearby A P dealer. Where's that? Phone (toll-free) 800-321-9668. And ask for the complete A P catalog, The Faster and Easier Book.

A P PRODUCTS INCORPORATED ✓ A22

Box 110 E • 72 Corwin Drive
Painesville, Ohio 44077
Tel. 216/354-2101
TWX: 810-425-2250

Faster and Easier is what we're all about.

ROM set. Once upgraded, your old PET will use the same memory locations as a new PET. Thus, most PEEK or POKEs used in old PET programs will not function properly and may even lock out your PET. You can modify your programs using my conversion chart starting on p. 72 of the July issue of *Microcomputing*. Since programs using machine-language subroutines probably won't work with the new ROM upgrade, what becomes of your old ROM set? Do you throw it away?

Small System Services, Inc. (900 Spring Garden St., Greensboro NC 27403), has what seems to be an excellent answer to this question: a small board that plugs into your ROM sockets. It comes completely assembled for $54.95 and is easy to hook up. Both your old and new ROMs plug into the board, and can be activated with the flip of a switch.

This board is called the Basic Switch. For $149.95, SSS will send it to you with the new ROMs already in the correct sockets, making it even easier for you, plus saving you $5 over purchasing the new ROMs directly from Commodore. A deluxe model will be available with an extra socket or two. You can plug the Programmer's Tool Kit or other software on ROM into these sockets. For more details, contact Small System Services.

(Special note: There are two different types of old ROMs. The new upgrade for the old 6540 set is available now. The new upgrade for the old 6316 set should be available in August. If you order the Basic Switch, be sure to specify your PET serial number and the type of old ROM in your PET so SSS can send you the correct version of their board. You should also know that the Basic Switch board doesn't connect to any of the PET expansion ports, which might be a problem for those with things already attached. Anyone upgrading his old PET directly with ROMs and not using the Basic Switch should save his old ROM set. Who knows—maybe someone will put out a board for the new PET allowing it to use the old ROMs.)

Program Protection Encore

Many software companies are concerned about illegal copying of their programs and are now attempting to protect their programs from being copied. Computhink has a method to protect disk programs with their disk system. I've also discovered an easy method with their system. Of course, I won't tell you how it's done, but any software marketer is welcome to contact me, and I can send an example. CMS Software has developed a method of protection using the PET disk system. NEECO also has worked on a protection method.

```
1 PT = PEEK(50003):SL = 537 - 393*PT:DL = 136 - 87*PT:POKE SL, DL
```

Example 1.

```
900 PT = PEEK(50003):SL = 537 - 393*PT:DL = 133 - 87*PT:POKE SL, DL
```

Example 2.

CMS Software

CMS Software is now offering several comprehensive business software packages for the PET with Commodore dual disk drives. They also are looking for specialized programs to use with their accounting packages. CMS will supply all the subroutines and pay royalties up to 60 percent. I was fortunate to receive a preliminary version of the CMS General Ledger. The diskette with documentation has an announced price of $295. Also announced at $295 each are Payroll with Cost Accounting and Accounts Payable/Accounts Receivable. Contact CMS Software for the name of your nearest dealer.

Software for Schools

Conduit (PO Box 388, Iowa City IA 52240) has 11 software packages in physics, biology and chemistry on cassette for the PET. These programs are described in detail in *Pipeline*, Vol. 4, No. 2. *Pipeline* is available free from Conduit. Simply send a postcard and ask to be placed on their mailing list.

Microphys Programs (2048 Ford St., Brooklyn NY 11229) has announced nearly 100 programs in various subjects, mainly for high schools. Subjects included are chemistry, physics, utilities, math and vocabulary. Several of these programs are also being offered by Commodore. Some of the programs are computer assisted instruction, while others are individualized instruction.

Programs from both Conduit and Microphys should prove useful in the classroom. Commodore also has announced programs for educators. Their programs include a series in elementary language arts written by Thorwald Esbensen. I hope to see these soon. I have seen other programs by Mr. Esbensen, and he does a wonderful job of making them both instructional and interesting for children.

Hardware Reset

International Technical Systems (Box 264, Woodbridge VA 22194) is selling a little device consisting of a push button, which mounts on the side of your PET, and two alligator clips that "hook" onto two locations inside the PET. It installs easily . . . one push of the button and your BYTES FREE message is back on the screen. This is useful if you are doing machine-language programming and you are "locked out." With this button, you won't have to turn your PET off and back on, which is hard on your monitor. A touch of the button resets the PET and clears all memory, and apparently the cassette buffers are left intact—a good side benefit.

Cassette Tape Sources

I've received favorable comments on the following companies' cassettes (please write with your notes on both good and bad cassettes for use with the PET):

Dr. Daley, 425 Grove Ave., Berrein Springs MI 49103.

AB Computers, PO Box 104, Montgomeryville PA 18944.

Computer Applications, Box 203, 446 Newbridge Ave., East Meadow NY 11554.

DAK Industries and Long Electronics have been mentioned, but I have no addresses to list.

Stop Key Disable

To disable the STOP key, your program will have to determine on which version of PET it is running. A PEEK at location 50003 will confirm whether it's a new or an old PET. A new PET will return a 1, while the old PET will return a 0. See my article on p. 72 of the July issue for a discussion on using this information. Briefly, the following line will allow you to use variable PT as the PET Type: 0 PT = PEEK(50003).

To disable the STOP key on the old PET, POKE 537,136; on the new PET, POKE 144,49. Micro Software Systems (PO Box 1442, Woodbridge VA 22193) was kind enough to supply the following modification of my one-line STOP disable (July issue). If executed more than once, my version could cause loss of control. Their version is shown in Example 1. To reenable the STOP key, use Example 2. If you previously assigned PT and SL, you need not repeat the first section of the reenable line. You could then simply use: 900 DL = 133-87*PT: POKE SL,DL.

These lines will operate on either old or new PETs without modification since the lines are "smart" and "know" which PET they are running on. There are possible side effects with this method, however. I believe it messes up the PET jiffy counter. If your program uses it, you may have to either change that section of your program or use another method of STOP disable.

Macro Assembler/Text Editor

This assembler from C. W. Moser (3239 Linda Dr., Winston-Salem NC 27106) is written entirely in machine language, not in BASIC as others for the PET often are. Thus you get fast and accurate assemblies. It is an 8K program, so you need at least some memory expansion for your 8K PET to be able to use this program. It features 26 commands, 22 pseudo ops and 36 error codes. Included are vectors and special commands for use with disk op-

(see PET-POURRI, page 190)

NEW PRODUCTS

Edited by Dennis Brisson

TRS-80 Minifloppy Disk Drive

Percom Data Co., 211 N. Kirby, Garland TX 75042, is advertising a TRS-80-compatible add-on minifloppy disk drive. You can purchase a single, double or triple disk-drive unit to expand your present system. You will need an expansion interface, TRSDOS diskette and TRSDOS manual to use the Percom drives. These same items are required to run the Tandy (Radio Shack) disk drives.

Instant Software, Inc. (ISI), purchased a triple drive unit, which arrived with three Wangco Model 82 disk drives (Pertec FD 200 disk drives are also used), a cabinet with power supplies, a new ribbon interconnect cable, a MICRODOS diskette and MICRODOS manual. Each disk drive is packed separately, and you have to assemble the units into the case. The only tool we needed was a straight-blade screwdriver. The disk drives were ready to be connected to the TRS-80 in about half an hour.

The best surprise came when we tried to run the system. All three disk drives functioned properly right from the start. We had a Tandy single drive that we sent back to Tandy three times before it functioned correctly. The Wangco drives are slower (40 millisecond track access time) than the Shugart drives used by Tandy, but the time difference is not a significant factor during operation. Data transfer to and from the disk drives has been solid and error free.

The MICRODOS is a simple, yet versatile, disk operating system for the TRS-80, but it is not compatible with TRSDOS. MICRODOS is composed of four programs: a menu, a simple disk file manager, disk utilities (which include backup, erase and copy) and the Percom notebook. One of the niceties is a merge command that will merge a routine from disk into a program in memory. This allows building an extensive subroutine file on disk and merging into the program. You write the subroutine once and then call it up anytime you want it.

ISI did have one problem with the Wangco drives. There is an interlock device in these drives that we were activating by finger to get the door closed. The disk drives are mounted on edge in the Percom housing unit. With the housing on its side, the disk drives are horizontal with their doors open. In this position the interlocking device started working properly. This particular door interlock requires proper orientation to function, but this in no way impairs the use of the drives. We overcame the problem of the sticking door interlock.

That the Percom system functioned properly the first time is recommendation enough, but couple that with MICRODOS and its low cost, and you have a good system for the money. Reader Service number P64.

Edward E. Ulmer
Technical Dept., ISI

Percom's TRS-80 triple drive unit.

Imagination Machine

The Imagination Machine is a personal computer and home entertainment center in one. It can be used for educational guidance, home and personal-budget management, as well as for leisure-time activities such as games of skill, chance, dexterity and multi-player competition.

Through a simple hookup with your existing TV set, you can start up your Imagination Machine, which is user programmable. At high resolution, the system's color graphics consist of 128 × 192 characters with up to eight color variations.

Standard features include 10K ROM and 9K RAM memory, typewriter-type keyboard, 53 keys with a unique shifted "BASIC Keywords" button, two game-style controllers, 32 characters × 16 lines screen format, six function built-in cassette tape deck, digital and audio information record/playback, built-in microphone jack, cartridge connector plus an internal operating system and BASIC interpreter.

The Imagination Machine has a built-in music synthesizer with a range of three octaves including sharps and flats. Expandable options include printers, floppy disks, word-processing capabilities and expandable RAM plus coupler modems.

APF Electronics, Inc., 444 Madison Ave., New York NY 10022. Reader Service number A103.

APF's Imagination Machine.

Apple II Business System.

Business System

The Apple II Business System combines all the advantages of a personal computer—portability, accessibility and affordability—with versatile business software. The configuration includes the Apple II Plus Computer, which is an upgrade of the Apple II first sold in 1977, 48K bytes of RAM memory, two disk drives, a video monitor, a printer and the Controller software package.

The system has an expanded version of built-in BASIC that makes it well suited for business problem solving. In addition, an Auto-Start ROM chip provides automatic disk loading as the system turns on, reset protection and easy screen editing. The Controller business software package, designed especially to meet the needs of smaller businesses, consists of three program modules:

In our July 1979 mention of Microsoft's M6800 BASIC (p. 16), we failed to state that the BASIC is licensed only on an OEM contract basis; it is not now directly available to end users.

The LVM-80.

Exidy's Video/Disk.

General Ledger, Accounts Receivable and Accounts Payable.

The Controller package includes a unique fail-safe feature of the data entry system that signals typing errors with an audible warning. The Controller automatically makes copies of data files for historical purposes in case of lost originals and automatically prints reports before the system will allow the user to close out the monthly books. The system costs $4995.

Apple Computer, Inc., 10260 Bandley Drive, Cupertino CA 95051. Reader Service number A28.

Graphics Digitizer

The Simple II is a new digitizer aimed at systems with graphics applications ranging from medical analysis to interactive graphics to engineering design. The Simple II is an 11 × 11 inch self-contained digitizer with 400 lines per inch resolution and 10 mil accuracy. X-Y output is variable up to 240 coordinate pairs per second.

The Simple II, from Talos Systems, Inc., 7419 E. Helms Dr., Scottsdale AZ 85260, will replace the Simple One as the firm's small low-priced graphics digitizer. The new unit will sell for $1065. The Simple II will not replace the 600 series digitizers the firm currently offers, but will serve where high resolution, backlighting or rear projection are not necessary. Reader Service number T70.

Digitized Audio Response System

The Votrax LVM-80 is a solid-state multiplexed audio response system that provides multi-line audio output and Touch-Tone input handling capability for microcomputers, minicomputers and large business mainframes. The vocabulary capacity of the LVM-80 is 512 seconds of speech, or approximately 1024 individually addressable messages. Memory capacity of the LVM-80 is 16K (EPROM) and can be erased and reused should the user require reprogramming of the vocabulary by Votrax.

The LVM-80 communications processor services up to 64 simultaneous telephone lines and, through a process of digitizing actual human speech, can "speak" words and phrases with a quality that is virtually indistinguishable from the original speaker.

Votrax, 500 Stephenson Highway, Troy MI 48084. Reader Service number V26.

Video Display Board

The Electric Window video display board is memory-resident, programmable and displays up to 24 80-character lines. Named the Electric Window because the generated display is like a window through which you can observe the memory space where characters are being input and manipulated, this product is convenient in all applications, but has particular value in text and word processing where the effects of editing and line justification are instantly visible.

Electric Window features include: two character generators—one for standard characters and one that can be programmed for special characters—dual intensity, highlighting alphanumeric display; display scrolling controlled by a single programmable register; descenders on lowercase letters for improved readability; programmable display positioning; and programmable interlaced or non-interlaced scan. The Electric Window is compatible with standard video monitors. The instruction manual supplied with the Electric Window includes a complete operating system listing. An optional PROM operating system is also available. The Electric Window costs $249.95.

Percom Data Company, Inc., 211 N. Kirby, Garland TX 75042. Reader Service number P62.

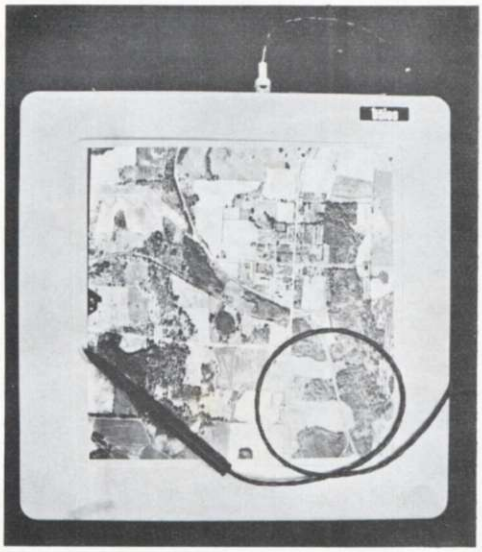
Video/Disk for the Sorcerer

A new Video/Disk peripheral attachment for the Sorcerer computer from Exidy Data Products, 390 Java Drive, Sunnyvale CA 94086, contains a 12 inch video display and dual mini-floppy disk drives with data storage capacity of 630,000 words. The swivel-based unit attaches directly to the Sorcerer's keyboard enclosure to form a desktop computer system no larger than a video terminal for small-business data processing. The disk drives utilize a unique precision metal positioner to achieve their storage capacity.

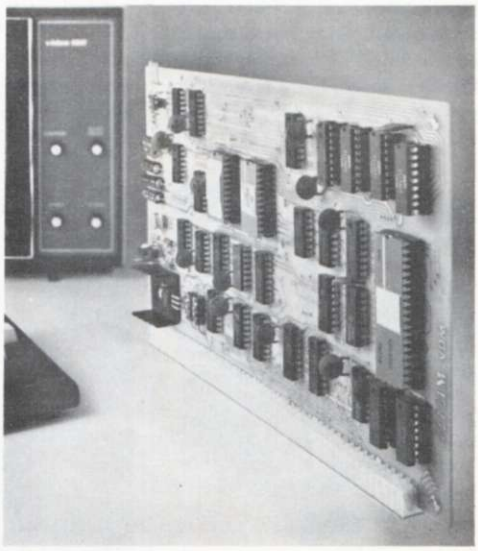
The high-performance video display utilizes a P31 phosphor for readability and a 20 MHz bandwidth for clear pictures to easily handle computer generation of 240 × 512 graphic resolution. Software included with the hardware consists of the popular CP/M operating system, Z-80 assembler, text editor, linking loader and Microsoft Disk Extended BASIC. The Exidy Video/Disk costs $2995. Reader Service number E44.

Microcomputer Modem

The Micromodem 100, a complete data communications system for S-100 microcomputers, combines on a single board functions



The Simple II.



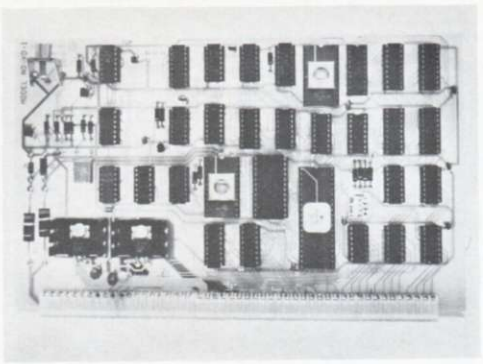
The Electric Window.

that formerly required a modem, an automatic calling unit and serial and parallel interfaces. The new board is a sophisticated computer-to-computer or terminal-to-computer modem for small computer systems.

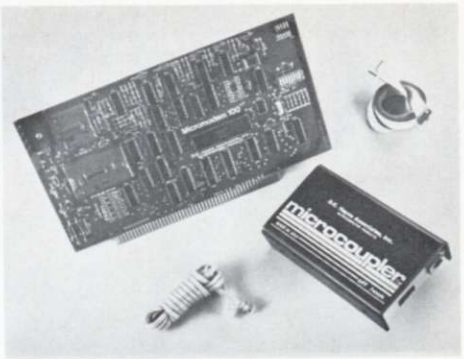
The Micromodem 100, fully S-100 bus compatible, including 16-bit machines and 4 MHz processors, converts digital data into analog signals for transmission over regular voice telephone lines. Because it is a Bell System 103 compatible modem, it can communicate with the most commonly used modems in North America. It operates at either of two software-selected baud rates—300 baud and a jumper-selectable speed from 45 to 300 baud—and is equipped with an FCC registered Microcoupler data access arrangement that feeds the signal into the telephone line through a modular connector provided by the local telephone company. Because of this direct connect feature, the Micromodem 100 can automatically answer the phone or dial a number. The Microcoupler provides direct access to the telephone system without the losses or distortions associated with acoustic couplers and without a telephone-company-supplied data access arrangement.

The Micromodem 100 serves as a convenient data communications tool between home and office, an automatic data collector from remote terminals, an intelligent terminal and allows implementation of remote software maintenance and customer support. It sends programs, letters and other data over the telephone and provides access to remote data bases. It is software compatible with the D. C. Hayes Associates 80-103A Data Communications Adapter, but offers improved performance and a direct connect and receive sensitivity of −50 dBm.

D. C. Hayes Associates, Inc., 16 Perimeter Park Drive, PO Box 9884, Atlanta GA 30319. Reader Service number H20.



The VDB-1 Smoke Writer.



The Micromodem 100 and Microcoupler.

Smoke Writer

The VDB-1 Smoke Writer is a fast SS-50 bus video display board featuring the MC6845 CRT controller chip to provide total control over display formats. The new board controls an 80 × 24 display with 32 graphic characters and has upper and lowercase characters with lowercase descenders. 128 character graphics are optional. Its programmable character set, a total of 128 characters, is in a 2K EPROM with 256 character 4K EPROM optional.

The Smoke Writer video board's other important features include: reduced intensity or reversed video, programmable display rate (10-5000 characters per second), protected fields, addressable cursor, 2K video display RAM accessible by the CPU as standard RAM memory, 128 bytes of scratchpad RAM and 1K EPROM for software drivers. Either a business program that needs protected fields or a cursor-based editor application will make good use of this video board. Price is $349.

Smoke Signal Broadcasting, 31336 Via Colinas, Westlake Village CA 91361. Reader Service number S46.

Battery-Operated Chess Computer

The Boris Diplomat is a full-featured, microcomputer-based chess computer that is designed for travel. It is compact (8 × 7 × 1½ inches), portable and battery operated (operates several hours on six AA batteries, not included, or on the ac adapter, included), with a built-in chessboard with pieces.

Designed with various operational strengths, the Diplomat will play at a level that will teach a child or will keep the attention of a master. As a teacher, it suggests moves for the unsure beginner. The position programmer allows more advanced players to set up special board positions to practice specific strategies. Beginners can use the position programmer to remove pieces for handicapping or for practicing specific positions. Price is $119.95.

Chafitz, Inc., 1055 First Street, Rockville MD 20850. Reader Service number C151.

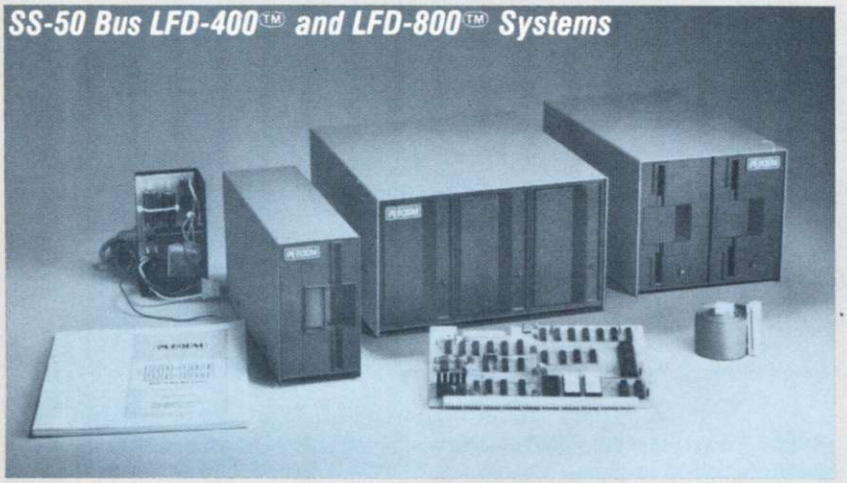
Computer and Terminal in One

NCE/CompuMart, PO Box 8610, Dept. P1, Ann Arbor MI 48107, combining the PET computer and NCE's own design in a direct spin-off from the PET technology, announces the TC 2001, a computer and a terminal in one unit at a price lower than a terminal alone. The TC 2001 functions as a full duplex, dumb terminal that can be used in conjunction with remote timesharing services, and also as an advanced personal computer. It incorporates one of today's most popular microprocessors, the MOS 6502, for fast and reliable operations. It is available with 8K, 16K of RAM memory. The ROM memory size is 14K. The TC 2001 uses BASIC language (Level II) and is fully expandable through IEEE, TTL parallel, second cas-

The Boris Diplomat.

The TC 2001.

Welcome to Percom's Wide World



SS-50 Bus LFD-400™ and LFD-800™ Systems

Each LFD mini-disk storage system includes:

- drives with integral power supplies in an enamel-finished enclosure
- a controller/interface with ROM operating system plus extra ROM capacity and 1K of RAM
- an interconnecting cable
- a comprehensive 80-page users manual

Low-Cost Mini-Disk Storage in the Size You Want.

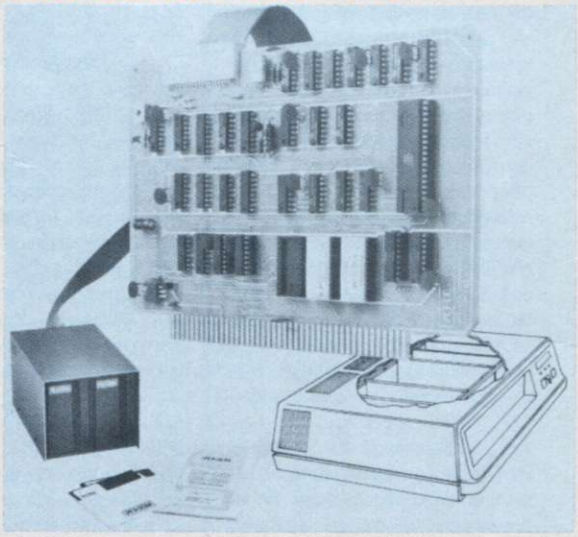
Percom LFD mini-disk drive systems are supplied complete and ready to plug in the moment they arrive. You don't even have to buy extra memory. Moreover, software support ranges from assembly language program development aids to high-speed disk operating systems and business application programs.

The LFD-400™ and -400EX™ systems and the LFD-800™ and -800EX™ systems are available in 1-, 2- and 3-drive configurations. The -400, -400EX drives store 102K bytes of formatted data on 40-track disks, and data may be stored on either surface of a disk. The -800, -800EX drives store 200K bytes of formatted data on 77-track disks.

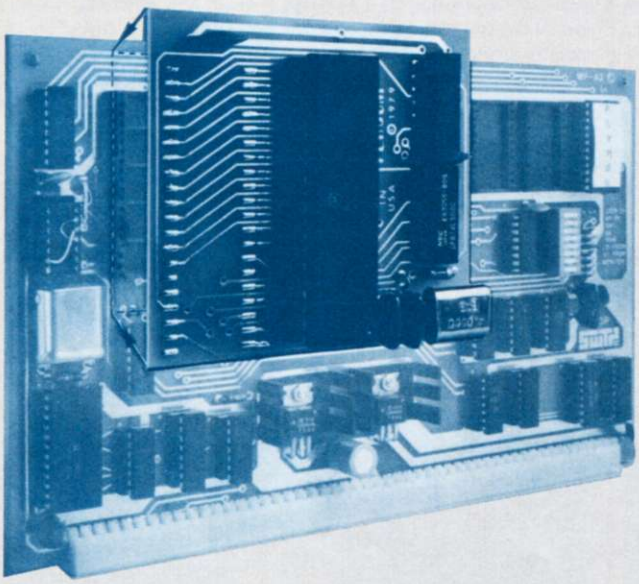
The LFD-1000™ systems (not pictured) have dual-drive units which store 800K bytes on-line. The LFD-1000™ controller accommodates two drive systems so that a user may have as much as 1.6M bytes on-line.

Mini-disk storage system prices:

MODEL	1-DRIVE SYSTEM	2-DRIVE SYSTEM	3-DRIVE SYSTEM
For the SS-50 Bus:			
LFD-400™	$ 599.95	$ 999.95	$1399.95
LFD-800™	895.95	1549.95	2195.95
For the EXORciser* Bus:			
LFD-400EX™	$ 649.95	$1049.95	$1449.95
LFD-800EX™	945.95	1599.95	2245.95
LFD-1000™	(dual) $2495.00	(quad) $4950.00	—



EXORciser* Bus LFD-400EX™ -800EX™ Systems



Upgrade to 6809 Computing Power. Only $69.95

Although designed with the SWTP 6800 owner in mind, this upgrade adapter may also be used with most other 6800 and 6802 MPUs. The adapter is supplied assembled and tested, and includes the 6809 IC, a crystal, other essential components and user instructions. Restore your original system by merely unplugging the adapter and a wire-jumpered DIP header, and re-inserting the original components. Also available for your upgraded system is PSYMON™ (Percom SYstem MONitor), the operating system for the Percom 6809 single-board computer. PSYMON™ on 2716 ROM costs only $69.95. On diskette (source and object files), only $29.95.

Data Terminal & Two-Cassette Interface — the CIS-30+

- Interface to data terminal and two cassette recorders with a unit only 1/10 the size of SWTP's AC-30.
- Select 30, 60 or 120 bytes per second cassette interfacing; 300, 600 or 1200 baud data terminal interfacing.
- Optional mod kits make CIS-30+ work with any microcomputer. (For MITS 680b, ask for Tech Memo TM-CIS-30+-09.)
- KC Standard/Bi-Phase-M (double frequency) cassette data encoding. Dependable self-clocking operation.
- Ordinary functions may be accomplished with 6800 Mikbug* monitor

Prices: Kit, $79.95; Assembled, $99.95. Prices include a comprehensive instruction manual. Also available: Test Cassette, Remote Control Kit (for program control of recorders), IC Socket Kit, MITS 680b mod documentation and Universal Adapter Kit (converts CIS-30+ for use with any computer).

of 6800 Microcomputing.

6800/6809 SOFTWARE

System Software

6800 Symbolic Assembler — Specify assembly options at time of assembly with this symbolic assembler. Source listing on diskette $29.95

Super BASIC — a 12K extended random access disk BASIC for the 6800 and 6809. Supports 44 commands and 31 functions. Interprets programs written in both SWTP 8K BASIC (versions 2.0, 2.2 & 2.3) and Super BASIC. Features: 9-digit BCD arithmetic, Print Using and Linput commands, and much more. Price $49.95

TOUCHUP™ — Modifies TSC's Text Editor and Text Processor for Percom mini-disk drive operation. Supplied on diskette complete with source listing $17.95

Operating Systems

INDEX™ — This easy-to-use disk-operating and file management system for 6800 microcomputers is fast. I/O devices are serviced by interrupt request. INDEX™ accesses peripherals the same as disk files — new devices may be added without changing the operating system. Other features: unlimited number of DOS commands may be added • over 60 system entry points • display only those files at or above user-specified file activity level • versions available for SWTP MF-68, Smoke's BFD-68 and Motorola's EXORciser*. Price $99.95

MINIDOS-PLUSX™ — An extension of the original MINIDOS™ for LFD-400™ mini-disk systems, MINIDOS-PLUSX™ manipulates files by six-character names. Supports up to 31 files. Resident commands include Initialize, Save, Allocate, Load, Files (directory list), Rename and Delete. Supplied on 2708 ROM with a minidiskette that includes transient utilities such as Copy, Backup, Create, Pack and Print Directory. Price ..... $34.95.

PSYMON™ — Percom SYstem MONitor for the Percom single-board/SS-50-bus-compatible 6809 computer accommodates user's application programs with any mix of peripherals **without** modifying programs. PSYMON™ also features character echoing to devices other than the communicating device, sophisticated register and memory dump routines and more. Price (on 2716 ROM) ..... $69.95.

WINDEX™ — Described in detail elsewhere on this page.

Business Programs

General Ledger — For 6800/6809 computers using Percom LFD mini-disk storage systems. Requires little or no knowledge of bookkeeping because the operator is prompted with non-technical questions during data entry. General Ledger updates account balances immediately — in real time, and will print financial statements immediately after journal entries. User selects and assigns own account numbers; tailors financial statements to firm's particular needs. Provides audit trail. Runs under Percom Super BASIC. Requires 24K bytes of RAM. Supplied on minidiskette with a comprehensive users manual. Price ..... $199.95.

FINDER™ — This general purpose data base manager is written in Percom Super BASIC. Works wth 6800/6809 computers using Percom LFD-400™ mini-disk drive storage systems. FINDER™ allows user to define and access records using his own terminology — customize file structures to specific needs. Basic commands are New, Change, Delete, Find and Pack. Add up to three user-defined commands. FINDER plus Super BASIC require 24K bytes of RAM. Supplied on minidiskette with a users manual. Price $99.95

Mailing List Processor — Powerful search, sort, create and update capability plus ability to store 700 addresses per minidiskette make this list processor efficient and easy to use. Runs under Percom Super BASIC. Requires 24K bytes of RAM. Supplied on minidiskette with a users manual. Price $99.95.

From the Software Works

Development and debugging programs for 6800 µCs on diskette:

Disassembler/Source Generator	$30.95
Reloc'tng Disas'mblr/Segmented Text Gen	$40.95
Disassembler/Trace	$25.95
Support Relocator Program	$25.95
Relocating Assembler/Linking Loader	$55.95
SmithBUG** (2716 EPROM)	$70.00

½-Price Special on Hemenway Software!

CP/68‡ disk operating system	$ 49.97
STRUBAL+‡ compiler	$124.97
EDIT68 text editor	$ 19.97
MACRO-Relocating Assembler	$ 39.97
Linkage Editor (LNKEDT68)	$ 24.97
Cross Reference utility	$ 14.97

™trademark of Percom Data Company, Inc.
* trademark of Motorola Corporation
‡Trademark of Hemenway Associates Company
**SmithBUG is a trademark of the Software Works Company

And 'looking into' is just what you do with the Electric Window™ as you peer right into memory space where characters are being input and manipulated. Display is memory-resident, programmable and generates up to 24 80-character lines.

Other features include:

- standard character generator plus provision for optional special character generator
- dual intensity, high-lighting alphanumeric display
- scrolling by a programmable register • programmable display positioning
- programmable interlaced or non-interlaced scan
- descenders on lower case letters • users manual with application instructions and listing of WINDEX™ driver.

The Electric Window.™ Worth Looking Into. $249.95

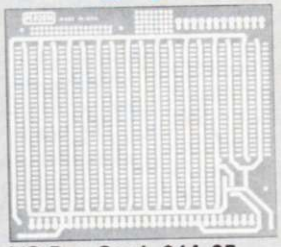
WINDEX™ is a fast video display driver program for the Electric Window™. WINDEX™ also features: program and keyboard control of character generators • displayable control characters — under program control • automatic scrolling • a driver routine for the parallel input keyboard feature of the Percom 6809 Single-Board Computer, the SBC/9™ • auto-linking to PSYMON™, the ROM operating system for the SBC/9™ • Prices: ROM version: $39.95; LFD-400™ compatible diskette (source and object files): $29.95.

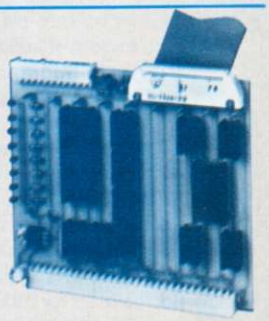
PDQ from PDC!

In the product development queue and available soon: **the SBC/9™** (Single-Board-Computer/6809) — stands alone as a control computer, but also compatible with the SS-50 bus for use as an MPU card. Includes PSYMON™ (Percom SYstem MONitor) in a 1K ROM and provides for additional 1K of ROM. Also includes 1K of RAM. Features: Super Port — provision for multi-address, 8-bit bidirectional data lines • an intelligent data bus for multi-level data bus decoding • an on-board 110-baud to 19.2 kbaud clock generator • extended address capability — to 16 megabytes — without disabling baud clock or adding hardware. And much more. Supplied with PSYMON™ and comprehensive users manual. Price ..... $199.95.

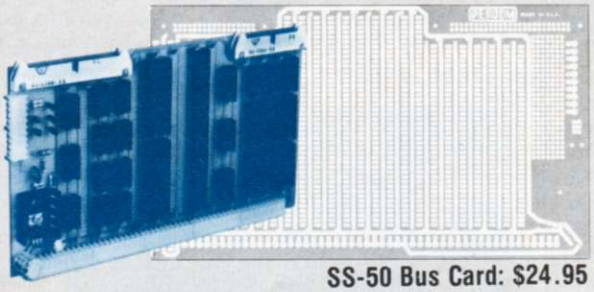
Full Feature Prototyping PC Boards

All of the features needed for rapid, straightforward circuit prototyping. Use 14-, 16-, 24- and 40-pin DIP sockets • SS-50 bus card accommodates 34- and 50-pin ribbon connectors on top edge, 10-pin Molex connector on side edge • I/O card accommodates 34-pin ribbon connector and 12-pin Molex on top edge





I/O Bus Card: $14.95



SS-50 Bus Card: $24.95

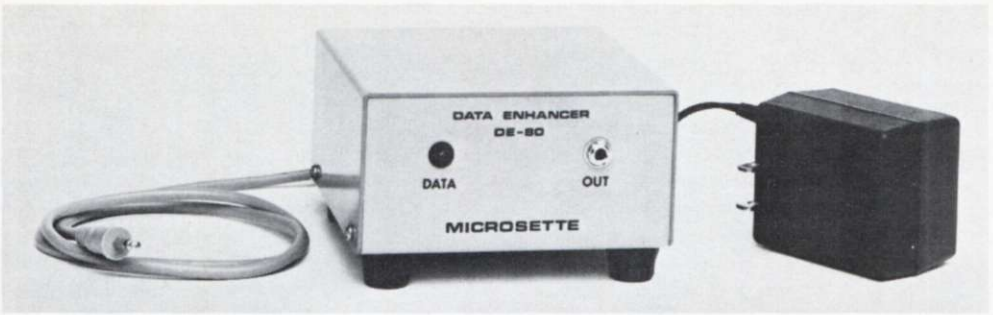
- I/O card is 1-¼ inches higher than SWTP I/O card • interdigitated power conductors • contacts for power regulators and distributed capacitance bypassing • use wire wrap, wiring pencil or solder wiring • tin-lead plating over 2-oz copper conductors wets quickly, solders easily • FR4-G10 epoxy-glass substrate.

To place an order or request additional literature call toll-free 1-800-527-1592. For technical information call (214) 272-3421. Orders may be paid by check, money order, COD or charged to a VISA or Master Charge account. Texas residents must add 5% sales tax.

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND, TEXAS 75042
(214) 272-3421



The DE-80.

sette and memory ports. Versions are available with either a full industry standard or calculator-type keyboard, which includes a cassette drive. Price of the unit with 8K RAM is $795.

As a terminal, the TC 2001 works at a 300 baud rate. It accepts RS-232 and outputs TTL voltage. The TC 2001 has both upper and lowercase characters, as well as graphics, on the 40 character by 25 line CRT display. A PET computer can be converted to obtain the same terminal capabilities by plugging a module, sold through NCE for $69, into the user port. NCE also offers an acoustic coupler and cable for both originate and answer mode. Reader Service number N7.

Data Enhancer

The DE-80 is an attachment between the Radio Shack TRS-80 and its cassette player to enable you to read marginally recorded data over a wide range of volume settings on the recorder. It will allow you to read either Level I or Level II cassettes from a volume setting anywhere between 4 and 10 on the Radio Shack CTR-type recorders, as compared to between 5 to 6½ if no enhancer is used in between. However, the DE-80 cannot make up for bad recordings or major tape dropouts where data has distorted playback waveforms, hum and variations in amplitude.

The DE-80 consists of a small 4 × 2½ inch box with only the read wire coming in and out of the box. It is inserted between the recorder and TRS-80 so there is no warranty violation from this improvement. The data enhancer also has an LED lamp that glows when data is being detected. The LED is useful for locating gaps between programs when more than one program is recorded on the cassette. Price is $45 with the power supply included.

Microsette Co., 777 Palomar Ave., Sunnyvale CA 94086. Reader Service number M67.

S-100 EPROM Programmer and 4K/8K EPROM Board

The PB1 programming board for 2708 and 5 V 2716 EPROMs is designed to meet all manufacturers' data sheet requirements for programming 2708s or 2716s. PB1 has two separate programming circuits so 2708 or 2716 (5 V) EPROMs can be programmed without modifying the board. Two textool programming sockets are provided for easy insertion and removal of EPROMs being programmed. Programming voltage is generated on-board by a dc-dc converter—no need for an external power supply. Programming sockets are DIP switch addressable to any 4K boundary. Special safety features to prevent accidental programming include an LED indicator and an on/off switch for programming voltage. Software listings are included for checking EPROM erasure, programming and verification.

The board also includes four additional on-board EPROMs that are independently addressable by DIP switch to any 4K (2708) or 8K (2716) boundary above 8000 hex. Unused sockets do not enable data bus drive. The board is fully buffered, and 0 to 4 clock cycle wait states are available. The PB1 has gold-plated edge contacts, plated through holes, low-profile sockets and factory prime parts; the PC board is made of FR-4 blue epoxy and is solder masked for ease of assembly. Price is $145, kit, and $219, assembled and tested.

Solid State Music, 2116 Walsh Avenue, Santa Clara CA 95050. Reader Service number S108.

D/A Converter Board

The Tecmar S-100 D/A board designed for applications requiring high-speed accurate digital-to-analog conversion, including real-time applications, supports four independent high-speed digital-to-analog converters (DACs) with associated latches. Each DAC operates completely independent of the rest. The DACs have a conversion time of three usec, which enables them to operate at maximum computer speed. A 12 bit latch drives the inputs of each DAC. Another four bit latch for each DAC holds the four new most significant bits waiting for the arrival of the new least significant byte. This allows the DAC to hold its previous value until an entire new word is presented to it.

All the latches are set to zero by reset. To modify the contents of a latch, and hence the output of a DAC, send two bytes to the device. The input is a 12 bit two's complement number. User-selectable output ranges of 0 to +5 V, 0 to +10 V, ±2.5 V, ±5 V and ±10 V are available. The board may be addressed as I/O ports or memory mapped. Only four lines of software are required to program these DACs. Price is $395, assembled and tested.

Tecmar, Inc., 23414 Greenlawn Ave., Cleveland OH 44122. Reader Service number T68.

Super Isolator

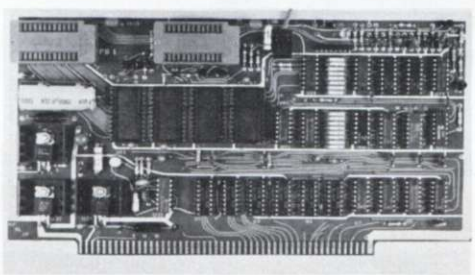
Now you can protect against extreme interference with the Model ISO-3 from Electronic Specialists, Inc., 171 S. Main Street, Natick MA 01760. The Model ISO-3 includes "superfiltering" for each of three 3-prong sockets. Heavy-duty spike and surge suppression is also included.

Intended for microprocessor installations in industrial or severe interference-prone environments, the Super Isolator also provides protection against damage from lightning and electrical machinery spikes and surges. It connects to the ac line with a standard 3-prong plug and can isolate and protect an 1875 Watt total load, with each socket capable of isolating a 1000 Watt load. Price is $72.95. Reader Service number E36.

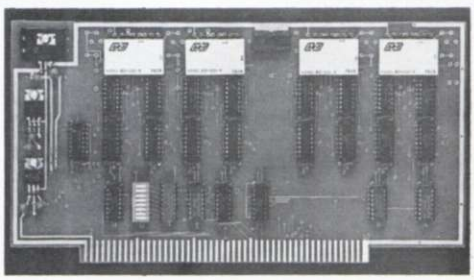
Level II BASIC Shorthand

TSHORT is a Level II BASIC shorthand that gives you single keystroke entry for 32 difficult-to-

The Model ISO-3.



PB1 EPROM programmer.



Tecmar's D/A board.

type, long and/or repetitive commands. TSHORT saves considerable programming time—up to 90 percent—and, by providing 100 percent accuracy, eliminates the programmer's biggest frustration—syntax errors. Immediately after each single keystroke entry, the entire command is spelled out in its normal format on the monitor and/or listing, i.e., there is no code to interpret.

TSHORT features a KUSTOM key, changeable anytime, for user-defined functions up to 64 characters of instruction, as well as a self-executing "Go To 10" function (line 10 may contain "RUN"). TSHORT includes the most popular and difficult Level II BASIC commands such as "RIGHT$(", "LEFT$(", "INPUT", "LPRINT", "CLOAD", "PRINTUSING", "GOSUB", etc.

TSHORT comes on cassette—one side for Level II and the other for DOS. What really makes TSHORT easy to use is the set of pressure-sensitive decals (white let-

(see PRODUCTS, page 191)

Keyboard with TSHORT commands.

LETTERS TO THE EDITOR

Overprotective?

I want to comment on the topic of protected programs. To put my remarks in context, I should mention that I am writing and have programs for sale and have had articles accepted to be published in current trade magazines. All this means is that I have some stake in the issue of copyright. No doubt I would be furious if someone copped my programs for his own and went into competition with me.

However, there is another side of the coin that should be considered before we all go running off to put a lock and chain on our programs. I've never met a program that I have not wanted to change to meet my own idiosyncracies. Even more important is that in this industry we have managed to develop a myriad of incompatible systems . . . even with the same names. For example, I could run a Cursor program designed for 8K memory on tape, but if that program was designed to use a cassette file system I'd sure want to convert it to disk files. I could not do that on a protected program.

What about the differences—not trivial—between the 8K PET, the CMC and the new 16/32K graphics version? Although a lot can be done to program around these differences, I doubt that we will be able to ever get to write a "universal" program. I can't.

In my mind there is no way to handle a good utilization program without allowing for modifications for a use peculiar to the user, and that's what we are writing for. One solution may be a method to protect segments of a program while allowing modifications of those areas that need to be customized.

Also consider for a moment what are the potential sources of program violation. I think that out-and-out plagiarism is minimal, and there are legal means of dealing with this. The major source or irritation lies with the "make a copy for me" syndrome between friends. This significantly reduces the market for the author of the program.

I believe that this problem can be minimized by good leadership in the user clubs and education of the potential users by editorials and letters such as this. I also think that this is the only solution because if someone is smart enough to protect a system, then there is always someone else that is just a little bit more clever who will find a way around the block. Micros are just too simple.

One final thought on the subject. The development of American literature is based upon the examples of the past. I feel that much the same idea applies in the area of good programming. One learns from concrete examples of successful programming on the market. The plethora of books on beginning programming and the dearth of any good material on advanced programming is readily evident. In my opinion, the widespread use of protected programs will adversely affect the quality of our industry in the long run.

In conclusion, I think the authors and publishers of programs should consider carefully the potential impact of protected programs prior to utilizing that technique. At least if one is marketing a protected program, then he should identify the fact so the buyer knows what he is getting.

Paul W. Sparks
Gales Ferry CT

"Whipping" POKE

When you append two or more programs (see Roger L. Pape's "Whip File Wipeouts," July 1979, p. 39), it may be necessary to add the step CLEAR:RESTORE in the command mode if your memory size disagrees with what it should be (add this step after re-POKEing the start of the program pointer to its default value).

In addition, it should be noted that the start of the program pointer will not always be 16548,233 and 16549,66. If TRSDOS is up, for example, it will be 186 (LSB) and 104 (MSB).

If you use someone else's disk system, or some newer or older version than 2.2, PEEK these locations without a program in the RAM and use those values. Keep in mind that to PEEK or POKE above 32767 you must use the formula: $-1*(65536-\text{desired address})$.

Robin L. Salmansohn
Abington PA

For the Birds

I personally feel we are wasting our time and energy, especially in printed form, for gadgets and games whether "hard or soft." I'm really upset about the enormous press coverage some *cheap* systems, i.e., TRS-80, are getting. I do not think a system should become a standard just because it has numbers. (Starlings, blackbirds and crows are not national birds.) I feel the microcomputer is headed for the "back door," and that's a "10-4." Let's at least make this electronic machine a little more useful to society. Let's start interfacing with the *real world!*

Tim Ziller
Grand Island, NB

Tim, I can see you're gonna just love my new magazine, 80-MICROCOMPUTING. Oh, I agree with the need for gadgets to permit our systems to reach out and accomplish tasks such as lawn watering, feeding the English sparrows, scratching our back, replacing our aging mailman. Go take a 10-100, good buddy.—Wayne.

– = – ; + = –

Greatly enjoyed "Murphy's Laws" (August 1979, p. 44).

Dzwonkiewicz's maxim: Negative thinking produces negative results. Positive thinking also produces negative results.

Dorsey F. Burk
Rancho Murieta CA

Meet You in Philly

I am involved in trying to develop, for profit, microcomputer applications for certain areas of the health-care industry, and I would like to establish communications with other individuals and organizations who are inter-

ested or involved in this particular area of microcomputer endeavors. Also, I am planning to attend "Personal Computing '79" in October in Philadelphia. I am hoping that I might get a chance to meet some of these people while I'm there. In fact, I'm proposing to get together during that weekend in Philadelphia such a group who, for now, I will call Health Care Microcomputer Entrepreneurs.

In addition to wanting to hear from anyone who is interested in this area of activity and who wants to exchange correspondence, I would like to receive specific indications of interest and intent from those individuals who would be interested in participating in such a meeting during that weekend (October 5-7).

Thomas H. Swalenberg
HCSMS
PO Box 13089
Columbus OH 43213

OK, your letter has been published. Now I expect you to keep Microcomputing *readers up to date on what develops.—Wayne.*

More on Morr

Many thanks to Dave Morr for his "TTY" program in the August 1979 issue of *Microcomputing* (p. 38). The program is *fast*, and it makes BASIC listings very readable thanks to the special symbols used in the printout. For those of you who are using a 60 wpm Baudot machine, as I am (Teletype KSR-32), a couple of changes that I've found successful might be of interest.

As mentioned in the article, location 7F93 must be changed; I found that 24 hex was the right value for my machine. I also found that, apparently due to the speed of the program, reversing the carriage-return and line-feed routines produced better results. This involves changing 7EFD to 02 and 7F02 to 08. If you prefer a 64-character-wide paper printout so that it will match the CRT of the TRS-80, then change 7EA1 to 3F. To use the cassette motor control relay for keying as shown in Fig. 2 of the article, simply change 7F88 to 14.

Before running the TRS-80 for a hard-copy printout, make sure that your TTY's carriage is returned and that the machine is in the LETTERS mode. The program does not print on the CRT screen when talking to the TTY, but this is very seldom needed anyway. Again, many thanks to Dave Morr for making available his well-thought-out program, and I hope that these tips may be of value to some readers.

David F. Miller
Niles IL

I want to commend you on your continuing publication of excellent applications articles for the TRS-80. An an amateur-radio operator, I was especially pleased with the article by David Morr on using the Baudot teleprinter for TRS-80 hard copy. The software ran nicely without modification, except that I got no automatic CR-LF after 72 characters.

However, when I first entered it and then typed LLIST or LPRINT, nothing would happen. Then I discovered that the driver addresses were not loaded at 16422 and 16423 decimal. I found that by adding the following BASIC statements:

```
POKE 16422,176
POKE 16423,1267
```

the program would function properly. Keep up the good work, and how about some RTTY-oriented Baudot input-output programs for the TRS-80?

PS. I used Rod Hallen's "Monitor" program from the June 1979 issue to load Morr's hex listing!

A. Podolsky, M.D.
Oak Park MI

It is not Baudot code, it is Murray code (see June 1978, p. 4). Those confounded Frenchmen have been taking credit for this for too long. Murray was an American, and our printers are and have been using Murray code and not Baudot. Got it? Other than that, thanks for the additions and notes; any others from other users of the system will be appreciated. The whole idea of this magazine is to allow computerists to help each other—Wayne.

Lawn Care by Computer

In "Machine-Language Monitors for the TRS-80" (August 1979, p. 114), author Robert Edmonds recommends that persons learn 8080 code first because it is a subset of the Z-80 code.

This is a common mistake; it arises because Zilog engineers wanted to preserve compatibility with existing software. However, the additions are not mere "enhancements." They comprise a different logical architecture that is more elegant and efficient than that of the 8080. Optimal Z-80 code is structurally different than 8080 code. This is primarily due to the index registers and bit operations; although there are lots of registers, memory operation capability has been significantly increased. While 8080 code must always shove data through the single accumulator, I find that fewer than half of my Z-80 subroutines ever access the accumulator at all. Because I learned 8080 first, it took me months to learn to program that way, but now my subroutines are faster and only half as long.

Furthermore, "Intel standard mnemonics" are very confusing, and when applied to Z-80 op codes, incomprehensible.

Come on, people, teaching 8080 code first is like teaching people to cut their lawn with a sickle before they can learn to use a riding lawnmower. It's a waste of time.

Richard Rodman
Vienna VA

So write an article telling us how to start writing Z-80 code, or at least give a good reference to read on it.—Wayne.

The Movie Was Great . . . .

I purchased Rodnay Zaks' *Programming the 6502* on the basis of your review in the July 1979 issue. Your reviewer mentioned: "On the negative side, Zaks' book contains a lot of typographical errors . . . most of them are trivial . . ." This is the understatement of the year! This book contains not only "lots," meaning hundreds, of trivial typos, but is generally illiterate in English construction, (Example, on page 46, "Exercise 3.1: Close now this book.")

However, more important is Zaks' lack of precise understanding of the 6502. I thought I could pick up some pointers on the 6502 from this book, but it is so rife with substantial errors that you must check every line if you really wish to use it. For example, on page 219 he has the code:

```
. . .
ADC #$00 RESTORE BIT
ROL CHAR SAVE BIT IN CHAR
. . .
```

After the ADC command, the carry will be clear. The ROL will always rotate a zero into CHAR so the bit won't be saved in CHAR!

If I buy a book, I prefer it to be mostly correct. Then I can learn from it and not always have to be on the defense against minor and major errors. With this book, I feel I was robbed! Almost anyone, novice or not, is going to have a hard time with it. Your reviews should provide such warnings!

Kendal T. Rogers
Fountain Valley CA

I'd like some more opinions on this since most of the comments on the book have been enthusiastic. Ken, aren't you blowing a few minor things out of proportion? There are some really lousy books on the market; let's go after these. Readers, if you get suckered into buying a bad book, please do write in so we can pass the word, but be sure you have it in perspective.—Wayne.

Plus and Minus

Let me take this opportunity to thank you for a great rag—one that is well worth the air-mail postage to New Zealand. Thank you for the advice to jump in and get my feet wet by buying some kind of micro even before I fully understood what was really going on.

I have been a computer "professional" for the last 11 years, but am now learning things about computers I never knew. It is reasonably difficult to stay up with the play at this distance, although we have an active club here in Auckland.

Some U.S. vendors are much more helpful than others. I'd like to put a plus by the names of Jade and Personal Software, and a minus by the names of Commodore and Ohio Scientific.

Jim Kennedy
Auckland New Zealand

Thanks

Since your publication of my letter (July 1979 Computer Clinic) asking for help in obtaining previous *Popular Electronics* articles on the COSMAC Elf there has been a tremendous response from your readers, and I have received all the information needed, and more, on the COSMAC Elf. Among your readers who sent copies of the Elf articles were Mr. Don Hartley, AR; Mr. John Lund, IN, Ms. Joyce Weisbecker, NJ; and Mr. Jeff Duntemann, NY. I would like to thank them for their enthusiastic help.

Moses Huang
Singapore

Food for Thought

Someday, every child will benefit from computer-assisted instruction.

Why not give your child the head start he deserves?

For the Apple*

Math Tutor I

The Math Tutor I package is designed to help younger children learn the concepts of mathematics. It uses the technique of positive, immediate reinforcement. When the child enters a correct answer, he or she is rewarded with the opportunity to move a spaceship, cast a magic spell, or rescue a man from a most unfortunate fate. By using this method of positive reinforcement, children will learn to associate mathematics with pleasant experiences, as opposed to the stark reality of numbers that has driven so many pupils to an early dislike of math. The program also provides the opportunity for parents to indulge in a unique learning experience with their children, while introducing them to the exciting world of microcomputing. Teachers can also use the program in a classroom setting to provide that extra incentive. Math Tutor I contains the following programs:

Hanging

A variation of the traditional game of Hangman. In Hanging, students must spare the graphic character's "life" by correctly answering the questions posed by the computer.

Spellbinder

The student is cast as a magician competing against another magician who's controlled by the computer. He must answer the questions about fractions correctly in order to cast magic clouds, fireballs, and other spells at his rival.

Whole Space

The student is given the opportunity to lead an attack on an evil planet. Every time a correct answer is given to a math problem posed by the computer, he can move his ship closer to his goal. Every wrong answer gives the enemy an opportunity to fire back.
Order No. 0073A $7.95

Note: The Hanging and Spellbinder programs require Applesoft II BASIC. The Whole Space program can run in Integer BASIC.

Math Tutor II

Math Tutor II is designed to complement the Math Tutor I package and introduces students to more advanced mathematical principles. Again, it uses the technique of positive reinforcement by offering rewards for correct answers—a principle long recognized as effective in helping students develop healthy attitudes towards study. To keep pace with individual pupil's development, all of the programs in this package may be used with any one of four levels of difficulty. Package includes:

Car Jump

Make the car jump over the buses. Every time your student correctly calculates the areas of various geometric planes, the number of buses in the line is increased by one.

Robot Duel

Each time a student correctly answers a question involving volumes of geometric solids, he's allowed one shot at the computer "robot." For every incorrect answer, his shield power drops, and the enemy robot is given the opportunity to shoot back.

Sub Attack

Students get to practice using percentages as they maneuver their sub into the harbor. A correct answer lets them fire their torpedoes at the enemy fleet.
Order No. 0098A $7.95

Note: All programs in this package require Applesoft II BASIC.

*Apple and Applesoft II are trademarks of Apple Computer Inc.
**TRS-80 is a registered trademark of Tandy Corp.

See page 71 for valuable coupon offer

For the TRS-80**

Teacher

This practical program allows you to input any number of questions and answers from the fields of science, languages, history—you name it. Using this data, the computer will prepare several types of tests, quiz students—providing up to three hints per question—even offer graphic rewards for younger children, all at the user's discretion. True or False tests may also be given.

Included with the Teacher package is a separate program, Teacher Data transfer, that allows several tests to be recorded sequentially on one cassette, thereby permitting the administration of whole lessons without the inconvenience of loading individual cassettes. Teacher is effective, practical, easy to use—perfect for teachers, parents, businessmen or anyone faced with learning a lot of material in the shortest possible time. Furnished with blank data cassette.
Order No. 0065R $9.95

If your local Instant Software dealer doesn't have the program you're looking for, you can order direct: Just use this order blank or call Toll-Free 1-800-258-5473.

Name _____

Address _____

City _____

State _____ Zip _____

☐ Check ☐ Money order

☐ VISA ☐ AMEX ☐ Master Charge

Card No. _____

Expiration Date _____

Signed _____ Date _____

Order your Instant Software today!

Quantity	Order No.	Unit Cost	Total Cost
		Handling	$1.00
		Total Order	

Instant Software Inc. Dept. 69J3
Peterborough, N.H. 03458 USA

As with all Instant Software programs, Math Tutor I & II and Teacher are sold complete with detailed instructions that take even the novice step by step, from loading and running the program, through suggestions for use and care of cassettes.

Instant Software™ Inc.

Peterborough, N.H. 03458 603-924-7296

Phillip Schuman
1627 Woodcutter
Wheaton IL 60187

Peter A. Stark
PO Box 209
Mt. Kisco NY 10549

Thoughts on the SWTP Computer System

Part 6 of this series finds a coauthor, Phil Schuman. In this installment, Pete and Phil take a close look at three of the six disks that are available for the SWTP computer system.

There are six disk systems available for the SWTP system at this time—three large floppy and three mini-floppy. The large ones (Midwest Scientific Instr., SWTP and Smoke Signal Broadcasting) are more expensive and quite possibly outside the reach of many. But the three minis are more reasonably priced, and are the subject of this review. They are:

The LFD-400 from Percom Data Company, 211 N. Kirby, Garland TX 75042, at $600 for a wired single-drive system or $1000 for a wired dual-drive system.

The BFD-68 from Smoke Signal Broadcasting, 31336 Via Colinas, Westlake Village CA 91361, at $800 for a single-drive wired, $1140 for a dual-drive system or $1480 for a triple-drive system. A less expensive single-drive system ($650), called the ABFD-68, comes without case or power supply.

The MF-68 from Southwest Technical Products Corp., 219 W. Rhapsody, San Antonio TX 78216, at $900 kit, $1000 wired for a two-drive system.

There are also several hybrids among these systems. Smoke Signal sells an MF-68-U kit, which includes the SSB controller board, to change an SWTP MF-68 system into an SSB-compatible system that will run SSB software. It costs $270. (With this kit, some drives, a case, power supply and cable, you could build your own SSB-compatible system.)

Peripheral Technology Associates, 3848 Hampton Dr., Columbus GA 31904, sells a disk controller, model FD-1, which is compatible with SWTP MF-68 software ($27 for a bare board, $110 for a wired controller). With this board you could change an SSB or Percom system into one compatible with SWTP MF-68 software. (Or, with your own drive, case, power supply and cable, you could build your own.)

Likewise, Percom sells their controller separately for $115, wired; using this you could modify an SSB or SWTP system into a Percom, or build your own.

The Micro Works, PO Box 1110, Del Mar CA 92014, offers a retrofit kit (model DM-85 at $40) for combining 5-inch mini and 8-inch regular floppy drives on the same SSB disk controller. It only works with later controller boards.

Disk System Hardware

A complete disk system consists of two parts—the hardware and the software. One is no good without the other. Moreover, there has to be good software support; the world's best hardware will do you no good without a good selection of software available. Let's talk about the hardware first.

Mini-floppy disk systems consist of several parts. First, there is the controller. This printed circuit board contains the electronics for controlling the disk drives and mounts inside the computer. The SSB and Percom boards are large and fit into the 50-pin bus in the SWTP system, next to memory or CPU boards. The SWTP controller is a smaller board that plugs into the 30-pin I/O bus. (If your system is already overloaded with boards and has no empty slots in one bus or the other, your decision is made right here.)

Then there is the drive. All of the commonly used drives, such as the Shugart SA-400, Wangco model 82, Pertec or whatever, are exact replacements for each other. They have the same mounting screws, the same connector and the same connections. You could mix two different drives in the same system, although there are good reasons not to do so. There are slight differences: Some drives allow you to flip a disk over and use the back, some may move the head from track to track a little faster and some may have more tracks.

The drive has to be mounted in a case and connected to a power supply. Since the controller board is powered by the mainframe, the power supply is only for the drive or drives. It mounts in the same case. There are small differences here among the three systems.

Percom's basic one-drive system comes in a small case for just one drive; it has a small supply only capable of powering one drive. A two-drive system has a larger case and two power supplies. Updating from a single-drive system to a two-drive system either results in two separate one-drive cases or requires an extra charge for a larger case.

The SWTP case comes with two drives and a power supply for two. (Expanding beyond two requires a second case and power supply.) It's not available with just one drive.

The SSB case and power supply are adequate for three drives, regardless of whether you buy the single, dual or triple system.

Connecting the controller

with the drives is a cable. Most systems use a 34-wire flat cable with crimped-on connectors at both ends. In multi-drive systems, one end of the cable simply has two or three connectors, separated by about 6 inches, which connect the same cable to all drives at the same time. With the exception of three wires in the cable, which select a particular drive, all other wires are shared among all three drives.

This makes the wiring simple, but has one disadvantage: When you read or write on one disk, all the drive motors go on together. This lack of independent motor control means that there is wear and tear on the unused drives and disks as well.

The disk drives and the Percom and SSB controllers use 34-pin two-sided card-edge connectors; the SWTP controller uses an AMP 87365-4 pin-type connector. Thus the same cable could be used for either Percom or SSB controllers, but another connector has to be installed if it is to be used on the SWTP controller. (By the way, the Radio Shack TRS-80 also uses a card-edge connector, so the same drives and cable could be used on SSB, Percom and TRS-80 systems.)

Software Requirements

Along with the hardware, you need some software. This is where the big differences really are among various systems. As far as the hardware is concerned, the three systems are all differently designed and built, but basically all work equally well. For most people, it's the software that sways them from one system to another.

Disk software consists of two parts—the disk operating system, or DOS, which is the entire package of software used to control the diskette, and the application programs, such as a BASIC interpreter, assembler or editor, which are used with the DOS.

The DOS is the essential program that controls everything the disk system does. It is usually broken down into three areas of responsibility: commands, the monitor, file management.

The commands are groups of programs that will perform selected functions, such as LOAD or SAVE, for the user. A basic set of commands is usually supplied with the system; some additional commands may be available at extra cost, and the user has the option to write his own commands in assembly language or add other software that will be treated as commands.

The monitor is that portion of the system that will maintain control, prompt for and check syntax of commands and interface to file-management routines. The monitor is usually a set of subroutines that may be accessed by the user (in assembly language) by way of a jump table, with an instruction such as JSR GETLINE or JSR ERRMSG.

The file-management portion is the set of routines and supporting work areas that access the diskette for reading and writing. In addition, a directory is maintained to ease in the searching of the diskette for specific files.

All of these components of the DOS may be resident in memory, but usually the commands and other low-usage routines are kept on the diskette and only loaded into memory as needed.

In addition, some DOS systems have other functions. There is sometimes a set of routines for terminal I/O, that is, for communicating with the terminal. For example, the SWTP DOS has its own routines for communicating with the terminal, while the SSB DOS uses the MIKBUG or SWTBUG ROM to perform the actual I/O.

When the computer system is first started, there must be some way of starting up the DOS. The SSB and SWTP systems keep all of the DOS on the diskette and require a mini loader to read the beginning portions of the DOS into memory. Once enough of the DOS is able to run, it loads the rest of itself into memory. The function of reading the beginning portions of DOS into memory is called booting, and the mini loader program is called the boot or bootstrap. In the SWTP system, the disk boot is in SWTBUG; in the SSB system, it is in ROM on the controller board; in the Percom system, the entire DOS is in ROM and does not have to be booted.

Another word to learn is IPL, or initial program load. This is the logical sequence of bringing a DOS up to a state of readiness. This may involve a booting, along with any other housekeeping functions, before the system is ready for commands.

Let's introduce another word that will be used later. A diskette has a *write-protect* notch along one edge. When this notch is left uncovered, that disk is unprotected and can be written on. But when a piece of tape is placed over the notch, the system senses the tape and will not write on that diskette. Hence the diskette is write-protected—protected from being written on.

With this introduction, let's look at each disk system one at a time.

SWTP MF-68

The SWTP MF-68 consists of a large cabinet with two drives, a power supply, controller board, connecting cable, a DOS and Disk BASIC. The DOS is called FLEX and is written for SWTP by Technical Systems Consultants, Inc. (TSC, Box 2574, W. Lafayette IN 47906). Although only two drives are supplied, two more can be added with an expansion kit.

The controller plugs into port 6 of the I/O bus in the mainframe. But it requires that a jumper be placed across two pins in I/O slot 5. SWTP suggests that you pull out the motherboard and solder the jumper underneath, but a much easier way is to simply plug a few extra female Molex connectors of the type used on the I/O boards into slot 5 and connect the pins on top. In any case, this jumper uses one of the user-defined (UD) lines on the bus and also makes it difficult to use port 5 for any other purpose (since its select line is being used by the disk). Thus the controller really uses up two I/O slots.

The controller uses the Western Digital 1771-01 floppy disk controller chip for disk formatting and is fairly uncomplicated, with just 13 ICs. The 1771-01 chip does all the work; most of the other ICs are used just for buffering inputs and outputs.

The bootstrap may be entered by hand, or you may use the disk bootstrap in SWTBUG. DOS occupies 4K of RAM at addresses 7000-7FFF; in addition, certain FLEX commands require a large buffer area, and, therefore, the system also requires at least 12K of RAM starting at address 0000. This area is used fairly seldom, however, and is otherwise available for programs (such as to hold the BASIC interpreter, for instance).

FLEX has its own sophisticated terminal I/O routines and, therefore, does not depend on the ROM monitor to provide this function. This means that any programs you write should use the FLEX I/O routines, rather than MIKBUG or SWTBUG I/O routines. This is somewhat of a problem; FLEX contains equivalents of the INEEE, OUTEEE or PDATA routines of the ROM monitor but does not have some of the others you might use, such as OUT2HS or BADDR. Hence, it may be difficult to work your way around it.

The system uses soft-sector diskettes, which have only one index hole and therefore have to be formatted before they can be used. This is done by the DOS, but requires several minutes for each disk.

The FLEX disk operating system is included in the price of the disk system (although TSC is now supplying a more versatile DOS called FLEX 2.0 on their own, at a price of $75).

FLEX commands reside on the diskette. The system comes with a basic set of commands. Also, TSC has written additional ones and sells them at extra cost, and commands can also be written by the user. Any executable file on the diskette may be used as a command; FLEX will attempt to load and start executing any command entered on the terminal. The command names, therefore, do not have any special naming conven-

tions. If, for example, you enter the word CLOAD, FLEX will look on the diskette for a program by that name to execute. If the program exists, it will be loaded and run. Hence, additional commands, beyond the functions that FLEX normally has, can be easily added to the system.

FLEX prompts the user with + + + and uses control-H or some user-defined character as the back space. The control-X is used to delete an entire line. A FLEX command called TTYSET controls these and other characteristics of the terminal. For instance, TTYSET controls the line length, whether the system echoes or does not echo characters being input, and so on. This command sets the total environment as far as the terminal is concerned. It even keeps track of the number of lines printed, so that output can be separated into pages on the printer. A pause option allows the output to stop at the end of a page; hence, users with CRT terminals can set up the paging and pause so that FLEX will fill up an entire screen with output and then wait until the user has read it before going on.

Since the SWTP disk system is normally supplied with two drives, it has provisions for using each drive for a different purpose. Using the ASN, or assign, command, you can assign one disk as a system disk and the other as a working disk. The system disk will have the DOS, its commands, BASIC interpreter, assembler or editor, while the working disk will have the data files used by these programs. This has the advantage that you can write-protect the system disk and protect it from accidental erasure by an errant program, while leaving the working disk unprotected so it can be written on.

FLEX Commands

Here is a short summary of FLEX commands supplied with the system. (There was an excellent review of this DOS by Mickey Ferguson in *Kilobaud*, October 1978, p. 72.) These commands are referred to as the Utility Command Set, or UCS, or sometimes just as disk utilities.

The first two commands are part of the DOS and sit in memory all the time:

GET—load a file into memory and return to FLEX.
MON—return to monitor (MIKBUG or SWTBUG).

The remaining commands reside on the disk and are called into memory only when needed:
APPEND—merge two or more files into one.
ASN—assign system and working disk drives.
BACKUP—create a mirror-image copy of a diskette.
BUILD—create a small text file.
CAT—list a directory of diskette on specified drive.
COPY—copy selected files from one disk to another.
COPYNEW—copy only those files that do not already exist on the second disk.
DELETE—delete a file from the disk, erase its name from the disk directory and free up the space formerly used by it.
EXEC—process a text file as a list of FLEX commands.
JUMP—start executing a program at a specified address.
LINK—set a pointer to inform the bootstrap program where to find the FLEX DOS on the diskette.
LIST—display the contents of a text or BASIC file.
NEWDISK—format a blank diskette with soft sectors.
P—steer the following command to use a parallel printer on port 7 instead of the terminal.
RENAME—change name of file.
SAVE—save a specified region of memory on the disk and optionally assign it an execution address so it can start executing when reloaded.
SAVE.LOW—used to save commands that will be loaded into the higher memory addresses used by FLEX.
TTYSET—set terminal characteristics and environment.
VERIFY—reread the disk after every write to check that there were no disk errors.
VERSION—display the version number of a specified command.
BASIC—start the BASIC interpreter.

In addition, FLEX has provisions for a file called STARTUP.TXT, which can contain a set of commands that will automatically be executed as soon as FLEX is booted. Hence, all the assigning of terminal output, system disk, working disk or other system parameters can be automatically done at start-up or IPL. This is specially intended for beginners or business users who don't want to be concerned with the nitty-gritty of booting the system and getting it going.

FLEX Features

FLEX maintains a directory on each disk of the files that are stored on it. File names consist of one to eight alphanumeric characters (the first of which must be a letter), followed by a period and a three-character extension, which specifies the kind of file it is. For instance, commands have the extension .CMD; text files end with .TXT; BASIC source programs have an extension of .BAS.

The diskette is divided into 35 tracks, and each track is further divided into 18 sectors. One track is reserved for the directory, with the remaining 34 tracks holding a total of 612 sectors. Each sector contains 124 data bytes, for a total usable disk capacity of 75,888 bytes. But since the DOS has to be on at least one disk, the first disk's capacity is closer to 50-60K bytes; only the second and following disks in a system will generally have the full amount usable.

Within a disk, sectors are allocated dynamically. That is, a given file may be spread out all over the disk, not necessarily in adjacent sectors. When starting with a blank disk, a disk-formatting process numbers all sectors and chains them together. That is, the directory holds the address of the first empty sector, that sector holds a pointer to the next empty sector, and so on down the line. Each sector is chained or linked to the next one. This is called a linked list, which links together all the empty sectors.

While DOS is formatting the disk, it is also checking each sector to make sure it has no defects and can store and read back data without errors. Any defective sector is automatically removed from the disk simply by not being linked in the chain of empty sectors. In this way, the DOS can use a disk even if it is not perfect.

When a file is placed onto an empty disk, it goes in the first empty available sectors, and then the pointer in the directory is updated to point to the first empty sector beyond that. As a disk is filled up, each file goes into the next empty sector in the chain.

At this point, there is one linked chain of all the empty sectors; in the directory, there is one pointer that points to the first of those empty sectors. Moreover, for each file on the disk, there is a directory entry that points to the first sector used by that file. Each sector, in turn, points to the next sector of that file. In both cases, the chain ends when the last pointer is 0000, indicating there is no more.

But when a file is deleted, its name is erased from the directory and its sectors are released, that is, their chain is added to the chained link of empty sectors. Now when a new file is added to the disk, it may start to use space that has been released from previous deletions.

In an intelligent DOS, new files will be placed into those areas of the disk that are big enough to hold them, if at all possible; in a dumber DOS, they will simply be placed into the first sectors in the chain of available ones. FLEX uses the latter technique, but either way, often there will be no single place big enough on the disk to hold a new file. In that case, it will get spread out among sectors all over the disk.

This has several advantages and disadvantages. Released space is immediately available to new files, rather than empty sectors being left here and there, unused. (For instance, Percom's disk doesn't do this; when a file is deleted on the Percom system, its sectors are attached to the file just before it on the disk. They cannot be used by other files until a PACK

program is run to pack all the used sectors together and close up all the empty holes.) Since files get spread out all over a disk, this evens out the wear on the disk surface, rather than concentrating it at the beginning of the disk.

On the other hand, when files are spread out all over the disk, more time is needed to read and write programs. For instance, it takes several times longer to load a large program from the SWTP disk than to load the same program from the Percom disk. Thus, it may be necessary to reorganize a FLEX disk occasionally, if just to speed up the access time.

FLEX handles disk I/O with its File Management System (FMS). These routines are resident in memory, accessed via a jump table, and provide the necessary functions to access a file.

BASIC and other application programs available for use with FLEX take care of reading and writing disk files automatically. But it is also possible to use the disk for programs written in assembly language. In this case, all diskette I/O functions are scheduled through a user-created work area called a File Control Block (FCB). An FCB is required for each file while it is being used. Several routines are available in FLEX to assist the user in properly loading the FCB with the correct information. Once it has been properly primed, it must be opened for I/O; after the I/O operations are complete, the FCB must be closed.

These functions are requested by placing the appropriate function code in the FCB and calling the FMS. The FMS routines, in turn, will handle the disk-related I/O and process a character at a time for the user. This is convenient for the programmer, since most non-disk programs operate with character-at-a-time I/O and are easily adapted to disk for this reason.

FLEX supports normal sequential access, along with random access. In addition, space compression and other special options are available, if requested in the FCB.

However, these are all functions not spelled out in the FLEX manual that comes with the disk system; you must purchase *DOS Advanced Programmer's Guide* for $5 to learn how to use them. This manual describes how to use disk I/O for assembly-language programs, but does not give any source listing of FLEX. Hence, you cannot go beyond the standard techniques as they are described in the manual. No source listings of FLEX or its commands are available.

FLEX is a versatile DOS and has an excellent collection of related programs available for it. The basic FLEX, which is supplied with the SWTP disk system at no charge, is called mini FLEX, but even so it is quite a package.

TSC is now selling several other versions of FLEX, including FLEX 2.0 for the SWTP disk ($75), FLEX 2.0 for the Smoke Signal Broadcasting mini-disk ($90) and FLEX 1.0 for full-size Smoke Signal disk. The new FLEX systems require 8K of memory from address A000 through BFFF (which means you may have to add one or two more memory boards), thus releasing all of the lower 32K of memory for user programs. (Since FLEX is then in a new area, old FLEX commands or programs will have to be changed to work with it.)

The new TSC FLEX versions differ from mini FLEX in a number of important ways. First, the new FLEX offers more storage per disk. Instead of 18 sectors per track with 124 data bytes each, the new FLEX formats the disk with ten sectors per track, with each track having 252 bytes of data. One track is reserved for the directory, so that the remaining 34 tracks hold 340 sectors, for a total of 85,680 bytes (instead of the 75,888 bytes in mini FLEX). New FLEX disks, therefore, have almost 13 percent greater storage capacity than mini FLEX disks, though this becomes significant only when each file is relatively large. This is because a small file of, say, 120 bytes takes up one sector in both systems; but in the 124-byte sectors of mini FLEX it

6800 SYSTEM SOFTWARE

Unmatched • Field Proven • Documented • Industry Wide

SDOS ™
A totally interrupt-driven (both disk and other peripherals) disk operating system, including type-ahead. Provides device independent, byte addressable random files. Supports any mixture of disk drives up to 2.5 BILLION bytes per drive. Disk files can grow dynamically to match application needs. Automatic, overlapped read-ahead on sequential files and LRU sector buffer pooling on random-access files optimize disk I/O. System utilities allow operator file manipulation, disk initialization, backup with wildcard file selection, and disk structure repair facilities to handle the infrequent but unavoidable disasters that occur in the real world. Turn-key application systems can be easily built, coupled with SD Business Basic. 242 pages of documentation.

IDB
A RAM or EPROM-based assembly language debugger. Provides single-step with register display, multiple real-time conditional breakpoints, memory dump, multiple data display and entry modes. Can be used to debug interrupt-driven code. 39 pages of documentation.

6800 Hardware supported:
Conrac Model 480 (AMI MDC) + ICOM floppy
WaveMate + Persci floppy (1771 + DMA)
Electronics Product Associates + ICOM floppy
Motorola EXORcisor + EXORdisk I or II
SWTP + mini or DMAF floppy (FLEX)
CMI 6800 + Winchester (16M) + Calcomp floppy (1771 + DMA)
MSI 6800 + FD-8 mini-floppy or 10M cartridge disk
Mizar Labs + double density Micropolis drives (1791 + DMA)
SSB Chieftain—mini or 8-inch floppy
Computer and Data Machines (England)

BUSINESS BASIC COMPILER
A super fast application oriented BASIC. 10 digit BCD for values to 100 million dollars with pennies. Random access to variable size, variable content records. Long, meaningful variable names, formatted output, IF-THEN-ELSE with multiple statements per line, and error-trapping make this BASIC extremely powerful. Compiled code, automatic integer optimization, and fast floating point make applications written in SD Basic run faster than on virtually any other microcomputer, and protect the source code of the application. 104 pages of documentation.

EDIT
A powerful and easy to use text editor with change, delete, insert, and remove commands. Automatic display of text or context changes, macro facilities for complex or repetitive editing. 44 pages of documentation.

ASM
A lovely 2 pass assembler with conditional assembly, long labels, symbol table dump and cross-reference, error cross-reference, extensive arithmetic and listing control. 103 pages of documentation.

Write for a free catalogue or contact the hardware manufacturer.

All SD software comes with a 1 year warranty.

INNOVATION IN SOFTWARE

SOFTWARE DYNAMICS
2111 W. Crescent, Suite G
Anaheim, CA 92804
(714) 635-4760

✓S112

wastes only four bytes, whereas in the 252-byte sectors of the new FLEX it wastes 132 bytes.

Since each sector contains twice as many bytes in FLEX as in mini FLEX, a given file will contain only half as many sectors. One reason for the slow speed of FLEX is that after every sector is written, FLEX waits for the disk to turn around and then reads the sector to make sure there are no errors. Since the new FLEX contains only half as many sectors in a given file, this process takes only half as long. Thus the new FLEX is faster than mini FLEX.

A major feature of the new FLEX is printer spooling. This is especially useful for running BASIC programs that require extensive—and slow—computation, combined with long printouts. Basically, spooling takes printer output and stores it back on the disk as a file, at high speed, instead of printing it. At a later time, while running another program whose output is also going on disk, the first program's output can be printed from the disk file at the same time. In this way, the program can continue running at top speed without waiting for the printer, while the printer can run continuously without waiting for the program. Essentially, two things go on at the same time. (The MP-T timer has to be installed in the computer to use the spooling feature.)

There is another use for spooling. It is possible to run a program, store its output on disk, and then examine that output on a CRT terminal before deciding whether to print it or not. This avoids a waste of paper—and time—if a program's results are not exactly what is wanted.

To implement printer spooling, the new FLEX has several new commands. The O command is used to specify that the output of the following program is to go to disk with an .OUT extension. After the output is on the disk, the PRINT command is used to instruct FLEX to put that program in a spooling queue to be printed when its turn comes. The status of programs in the queue can be checked at any time with a QCHEK command. QCHEK can also be used to delete individual files from the queue, or an XOUT command can delete all files that have an .OUT extension.

Printer spooling is a technique commonly used in large computer installations, and this is a first in the microcomputer area. But it also introduces some problems. For instance, if program output is going to a disk, what happens if the disk becomes full? Furthermore, if the program whose output is going on disk also happens to be using the same disk for files, the running time may be appreciably lengthened by the need for the head to move from track to track.

Since the print queue may contain several different files to be printed, there will often be a need to stop the printout between files to change to a different kind of paper, move to the top of a new page or make some other adjustments. Unfortunately, FLEX doesn't appear to have a convenient way of inserting a wait into its queue.

If there is a lot of printing to be done, the disk may be on for a long time during the printing; because of the design of the SWTP controller, the motor will run continuously during an extended print session. During an hour of printing, the disk will make 18,000 revolutions; most disk and drive manufacturers specify disk life as two or three million revolutions, but 18,000 revolutions in one hour is still a large number.

Finally, printer spooling would be a much more valuable feature if there were some provision for a second printer. (Incidentally, it is designed for a parallel printer in port 7, not for the terminal.)

(Percom's INDEX DOS takes another approach. Program output can be routed to the disk, as in the new FLEX, but its printing at a later time cannot be overlapped with other processing; it would have to be handled as a separate job all by itself. This provides the option of examining output before printing, but doesn't save the time that FLEX does. Instead, INDEX allows the printing to be overlapped with the main program itself, by using a buffer area in memory and interrupts to allow the printer to run at the same time as the computer processes data. This is another approach to the problem; FLEX may work out better in some cases, INDEX may be better in others.)

The new FLEX differs from mini FLEX in several other ways. For example, a disk is assigned a name when it is first initialized (though there is no mention in the manual of the name ever being used for anything). Files can be protected from being deleted, renamed or written over, which is an important feature. If you want to keep a file secret, you can also prevent it from being listed in a catalog printout. Some other minor differences are that some disk-resident commands, such as BACKUP, COPYNEW or MEMTEST, don't come with the new FLEX, or that if a file is not found on one of the drives, FLEX will look at the other drive.

What about other software? With mini FLEX, as part of the system purchase price, you get SWTP Disk BASIC Version 3.0. This is essentially SWTP 8K BASIC, which has been modified to work with the disk. The

Listing 1.

```
                        NAM  FLEX-TSC EDITOR
                        OPT  TAP,NOG
                    * ***************************
                    * * FLEX  TSC/EDITOR       *
                    * ***************************
                    * ***** FLEX-FMS FUNCTION CODES *****
0000                IO     EQU  0              CALL FOR I/O
0001                READ   EQU  1              OPEN FOR READ
0002                WRITE  EQU  2              OPEN FOR WRITE
0004                CLOSE  EQU  4              CLOSE FILE
                    * ***** FCB LAYOUT *****
0000                XFC    EQU  0              FCB+0  FUNCTION CODE
0001                XES    EQU  1              FCB+1  ERROR STATUS
0003                XUN    EQU  3              FCB+3  DRIVE NUMBER
0004                XFN    EQU  4              FCB+4-11  FILE NAME
000C                XEN    EQU  12             FCB+12-14  EXTENSION NAME
0001                TXT    EQU  1                   'TXT' FORMAT
003B                XCM    EQU  59             FCB+59  COMPRESS FLAG
0000                XON    EQU  0                    COMPRESSION ON
00FF                XOFF   EQU  $FF                  COMPRESSION OFF
                    * ***** FLEX-FMS ENTRY POINTS *****
7803                FMSCLS EQU  $7803          FMS CLOSE
7806                FMS    EQU  $7806          FMS CALL
                    * ***** FLEX ROUTINES *****
7103                WARMS  EQU  $7103          FLEX WARM START
7115                INBUFF EQU  $7115          INPUT DATA INTO LINE BUFFER
7127                GETFIL EQU  $7127          LOAD FCB FROM LINE BUFFER
712D                SETEXT EQU  $712D          SET FILE EXTENSION
713C                RPTERR EQU  $713C          REPORT DISK ERRORS
7118                PSTRNG EQU  $7118          PRINT STRING END IN $04
711E                PCRLF  EQU  $711E          PRINT CR AND LINE FEED
                    * ***** OPEN FILE FOR READ *****
1491                       ORG  $1491
1491 CE 15 0E       OR     LDX  #FCB           POINT TO OUR FCB
1494 86 01                 LDA  A #READ          CODE FOR OPEN READ
1496 A7 00                 STA  A XFC,X          STORE FUNCTION CODE INTO FCB
1498 BD 78 06              JSR  FMS            CALL FLEX FMS
149B 26 59                 BNE  ERR            BRANCH IF OPEN ERROR
149D 86 00                 LDA  A #IO            CODE FOR I/O PROCESSING
149F A7 00                 STA  A XFC,X          STORE FUNCTION CODE INTO FCB
14A1 39                    RTS
                    * ***** OPEN FILE FOR WRITE *****
14A2 CE 15 0E       OW     LDX  #FCB           POINT TO OUR FCB
14A5 86 02                 LDA  A #WRITE         CODE FOR OPEN WRITE
14A7 A7 00                 STA  A XFC,X          STORE FUNCTION CODE INTO FCB
14A9 BD 78 06              JSR  FMS            CALL FLEX FMS
14AC 26 48                 BNE  ERR            BRANCH IF OPEN ERROR
14AE 86 00                 LDA  A #IO            CODE FOR I/O PROCESSING
14B0 A7 00                 STA  A XFC,X          STORE FUNCTION CODE INTO FCB
14B2 39                    RTS
                    * ***** ASK TERMINAL FOR FILE NAME *****
14B3 BD 71 1E       ASK    JSR  PCRLF          PRINT CR AND LINE FEED
14B6 CE 15 CE              LDX  #ASKMSG        POINT TO OUR ?? MESSAGE
14B9 BD 71 18              JSR  PSTRNG         FLEX STRING OUTPUT RTN
14BC BD 71 15              JSR  INBUFF         INPUT RESPONSE INTO BUFFER
14BF CE 15 0E              LDX  #FCB           POINT TO OUR FCB
14C2 BD 71 27              JSR  GETFIL         LOAD FCB WITH DATA FROM BUFFER
14C5 25 09                 BCS  SERR           BRANCH IF SYNTAX ERROR
14C7 CE 15 0E              LDX  #FCB           POINT TO OUR FCB
14CA 86 01                 LDA  A #TXT           CODE FOR 'TXT'
14CC BD 71 2D              JSR  SETEXT         SET EXTENSION NAME
14CF 39                    RTS
14D0 BD 71 1E       SERR   JSR  PCRLF          PRINT CR AND LINE FEED
14D3 CE 15 D7              LDX  #SYNTAX        POINT TO ERR MSG
14D6 BD 71 18              JSR  PSTRNG         OUTPUT SYNTAX ERR MSG
14D9 20 D8                 BRA  ASK            GO ASK FOR IT AGAIN
                    * ***** READ/WRITE SUBROUTINE *****
14DB FF 15 E5       RW     STX  XSAVE          SAVE X-REG
14DE F7 15 E4              STAB BSAVE          SAVE B-ACCUM
```

SAVE, LOAD and APPEND commands of BASIC allow saving and loading BASIC programs on the disk; cassette commands are now TSAVE, TLOAD and TAPPEND.

In addition, BASIC allows sequential data files on the disk. A disk file is opened with an OPEN command, which ties a BASIC file number with a specific named file on the disk. Subsequent READ, WRITE and RESTORE statements read or write on the file and can reset it to the beginning, in a sequential fashion as if the data were going to tape. When data is done, you close the file. This BASIC does not allow random files, a big limitation to some users.

Having written FLEX, TSC is the biggest force behind other systems and application programs available for it. They have an excellent text editor ($31.50 for the disk version) for preparing text and program files, an assembler ($31.50), a text processor ($40), a debug package ($43), a package of disk utilities ($100), a disk sort-merge package ($75) and even a super-fast BASIC interpreter ($50), which is claimed to be the fastest BASIC on any 8-bit micro yet. But it only has sequential files, not random files.

```
14E1 CE 15 0E          LDX  #FCB          POINT TO OUR FCB
14E4 BD 78 06          JSR  FMS           CALL FLEX FMS
14E7 26 07             BNE  RERR          CHECK FOR I/O ERROR, OR 'EOF'
14E9 FE 15 E5          LDX  XSAVE         RESTORE X-REG
14EC F6 15 E4          LDA  B BSAVE         RESTORE B-ACCUM
14EF 39                RTS
                * ***** COMMON ERROR REPORTER *****
14F0 A6 01     RERR    LDA  A XES,X         RETRIEVE ERROR CODE
14F2 81 08             CMP  A #08           CHECK FOR 'EOF'  CODE=08
14F4 27 09             BEQ  EOF           IF FOUND, EOF-- ELSE ERROR
14F6 BD 71 3C  ERR     JSR  RPTERR        PRINT DISK ERROR CODE
14F9 BD 78 03          JSR  FMSCLS        CLOSE ALL OPEN FILES
14FC 7E 04 41          JMP  EDIT          GO BACK TO EDITOR PROGRAM
0441           EDIT    EQU  $0441         PRINT '?' AND EDITOR PROMPT
                * ***** READ 'EOF' ROUTINE *****
1458           READ25  EQU  $1458         RETURN TO PROCESS READ
1464           READ4   EQU  $1464         EOF HAS BEEN READ
14FF 86 04     EOF     LDA  A #CLOSE        CODE FOR CLOSE FILE
1501 A7 00             STA  A XFC,X         STORE FUNCTION CODE INTO FCB
1503 BD 78 06          JSR  FMS           CALL FLEX FMS
1506 26 EE             BNE  ERR           BRANCH IF ERROR ON CLOSE
1508 FE 15 E5          LDX  XSAVE         RESTORE X-REG
150B 7E 14 64          JMP  READ4         CONTINUE WITH 'EOF' PROCESSING
                * ***** CONSTANTS AND TERMINAL MESSAGES *****
150E           FCB     RMB  192           192 BYTE BLOCK FOR OUR FCB
15CE 46        ASKMSG  FCC  'FILE?  '
15D6 04                FCB  4
15D7 53        SYNTAX  FCC  'SYNTAX ERROR'
15E3 04                FCB  4
15E4           BSAVE   RMB  1             SAVE B-ACCUM
15E5           XSAVE   RMB  2             SAVE X-REG
15E7 0D                FCB  $0D           SET BEGINNING OF EDITOR BUFFER
15E8           BUFFER  EQU  *             USER BUFFER AREA
                * ***** OVERLAY OLD CODE, AND ADD PATCHES *****
020C                   ORG  $020C         I/O VECTORS
020C 7E 14 DB          JMP  RW            READ ROUTINE
020F 7E 14 DB          JMP  RW            WRITE ROUTINE
0212 4F FF             FDB  $4FFF         **** SET MEMORY LIMIT ****
                *
0272                   ORG  $0272         COMMAND TABLE
0272 04 41             FDB  $0441         'GAP' IS NOW INVALID
                *
0358                   ORG  $0358
0358 CE 15 E8          LDX  #BUFFER       POINT TO OUR NEW EDITOR BUFFER
                *
098B                   ORG  $098B         'EXIT' COMMAND
098B 7E 71 03          JMP  WARMS         GO BACK TO FLEX
                *
13D3                   ORG  $13D3         'WRITE' COMMAND
13D3 BD 14 B3          JSR  ASK           ASK FOR OUTPUT FILE NAME
13D6 BD 14 A2          JSR  OW            OPEN FILE FOR OUTPUT
13D9 01                NOP
                *
13F5                   ORG  $13F5         PROCESS WRITE EOF
13F5 CE 15 0E          LDX  #FCB          POINT TO OUR FCB
13F8 86 04             LDA  A #CLOSE        CODE FOR CLOSE FILE
13FA A7 00             STA  A XFC,X         STORE FUNCTION CODE INTO FCB
13FC BD 78 06          JSR  FMS           CALL FLEX FMS
13FF 27 03             BEQ  X2            IF OK, RETURN
1401 7E 14 F6          JMP  ERR             ELSE REPORT ERROR
1404 39        X2      RTS
                *
142D                   ORG  $142D         'READ' COMMAND
142D BD 14 B3          JSR  ASK           ASK FOR INPUT FILE NAME
1430 BD 14 91          JSR  OR            OPEN FILE FOR INPUT
1433 01                NOP
                *
146F                   ORG  $146F         DEVICE CONTROL
146F 01                FCB  1,1,1         3 NOP'S FOR DEVICE CONTROL
                *
                       END
```

The new FLEX 2.0 includes the text editor, assembler, advanced programmer's manual and documentation as part of the price, so that if you plan to purchase these, the improved FLEX package is a better deal. On the other hand, programs that run under mini FLEX require changes to make them run under the new FLEX; in some cases major changes may be needed. Furthermore, the new FLEX price does not include a BASIC interpreter; the new TSC super-fast BASIC in the new FLEX 2.0 form costs $55 (and includes random disk files).

Other FLEX-compatible software is available from Computerware Software Services (830 First Street, Encinitas CA 92024) and includes a PILOT interpreter, and various games and utility programs.

Ed Smith's Software Works (PO Box 339, Redondo Beach CA 90277) also has FLEX software, including a macro-assembler, disassembler, trace/debug package and others. A PILOT interpreter is also available from SWTP.

A BASIC renumbering program for FLEX BASIC version 3.0 is available from Lehigh Computer Works ($10, 1132-2 Tilghman, Allentown PA 18102).

Microware Systems Corp. (PO Box 4865, Des Moines IA 50304) also sells mini FLEX software, including their A/BASIC compiler ($150) and related interpreter and debugging software, a LIST interpreter, as well as a chess program and Eliza, a fascinating computerized psychiatrist program.

Advanced FLEX Programming Example

An example of using FLEX FMS and monitor routines is shown in Listing 1 for those interested in assembly-language programming of disk operations. This listing shows the patches to the cassette version of TSC's text editor, allowing it to run under FLEX (TSC sells a FLEX version of the Text Editor, but this version was adapted by Phil for users who already have the cassette version).

The beginning of the listing shows the entry points into the FLEX routines used, along with information concerning the FCB. The next segments of software show the open file, close file, prompt for file name and common I/O routines. These new segments are added to the editor software and referenced by some patches to the editor pointers. At the end of the listing are the patches that reset the pointers within the editor to reflect the newly added disk software.

A certain sequence of events must take place to properly access a disk file. Before the FCB is opened, it must be loaded with the correct information. The proper file name, with optional extension and drive number, must be set into the FCB. This was done using the FLEX INBUFF routine to accept data from the terminal, placing it in the line buffer. The FLEX routine GETFIL is then used to parse the line and load the FCB with the data that it finds in the line buffer. The SETEXT routine is then used to specify an extension of TXT. The file is now capable of being opened by a call to FMS. After the file is processed, it must be closed by a call to FMS.

Once the file is closed, the FCB may be used for any I/O to any file. In this case, only one FCB is used for both reading and writing of the edit buffer.

SSB DOS-68

Like the SWTP system, the SSB system consists of a cabinet, power supply, controller board, cable, a DOS called DOS-68 and Disk File BASIC. The cabinet and power supply can handle up to three drives, and the system can be bought in one-, two- or three-drive versions.

Like the SWTP disk system, the SSB system also uses soft-sectored disks (though their format is not compatible, and one system cannot read disks made on the other). The controller also uses the 1771-01 floppy disk controller IC, except that the controller in this case is a large board that plugs into the 50-pin bus, like a memory or CPU board. Hence, it does not use up I/O slots.

The hardware design of the controller is quite different from that of the SWTP controller. The SWTP controller relies on a bootstrap program in SWTBUG (which does not always work well the first time); the SSB controller has the bootstrap program and some other I/O routines in a 512-byte ROM on the controller card; so it can also be used with other monitors.

The controller design is interesting in other ways too. In a typical SWTP system, I/O ports 0 through 7 use the 32 addresses from 8000 through 801F. Addresses 8020 through 803F are left unused for those users who want to add a second motherboard to provide eight more I/O ports. But because of incomplete I/O address decoding, the addresses from 8000 through 801F are also decoded as 8040 through 805F, 8080 through 809F and so on, all the way up through 8FFF (or 9FFF in the older motherboard). The SSB controller interleaves its ROM bootstrap and I/O routines into the empty spaces between the I/O ports. That is, there are 32 bytes between 8020 and 802F, another 32 bytes from 8060 through 807F and so on, all the way up to 83FF. This is a novel way to sneak in some ROM into the system without interfering in any way with existing memory or I/O (unless you want to expand to a second motherboard and eight more I/O ports). The bootstrap program starts at address 8020, and so a simple jump to that address boots the disk.

The controller also uses a PIA at addresses 9FFC-9FFF for I/O; hence, you cannot have memory in the 9000-9FFF area, but this is not much of a limitation.

DOS-68 requires 4K of RAM memory at address 7000-7FFF but is available in optional versions that use 4K starting at addresses A000 or D000. A somewhat longer version of DOS-68 that supports random files is also available and requires 6K of memory at 6800-7FFF (optional forms are available at higher addresses).

Like FLEX, DOS-68 maintains a directory on the disk and chains together both used and empty sectors. File space is allocated dynamically, and free or available sectors are allocated as needed, so the disk need not be repacked to combine empty sectors (unless it is desired to speed up the system).

DOS-68 has a set of commands that are part of DOS and always in memory and an additional set of disk-resident commands (called transients) that reside on the disk and are loaded into memory only as needed. These include four memory-resident commands:

CLOSE—close all open disk files.

EXIT—return to the monitor.

GET—load file into memory and return to DOS.

RUN—load file into memory and execute it if applicable.

It also has the following disk-resident commands:

LIST—list the directory of a specified diskette.

SAVE—save a specified region of memory and an optional starting address.

GETH—load disk files created in MIKBUG punch format (S113 . . .) from assembler or patches.

DELETE—remove an entry from the directory and release the space allocated to that file.

RENAME—change the name of a file.

APPEND—combine two files into one.

PRINT—display the contents of a file.

COPY—copy a file from one disk to another.

SDC—single-disk copy for single-disk systems.

LINK—set a pointer to inform the bootstrap program where the DOS is stored on the disk.

INSTAL—rename a user file to a command file.

REMOVE—rename a command file into a user file.

FIND—display the starting and ending memory addresses of a binary program file.

VIEW—display a text file, adding cursor control as needed.

FORMAT—initialize a blank cassette.

ASYS—assign a system diskette location.

BASIC6—start the BASIC interpreter.

The SSB disk system comes with a Disk File BASIC. The interesting point is that, although Smoke Signal Broadcasting doesn't say so, this is Computerware Software Services' Super BASIC. It has every feature that we have gotten used to with SWTP cassette BASICs, plus a few new ones: a STRING command allows the string length to be changed; HOME issues a home-up and erase command for CRT terminals; SKIP skips lines on the terminal; WAIT inserts a time delay; RJUST allows neater printout of columns of figures by allowing decimal points to be lined up.

The BASIC disk commands include FLIST for listing the files on the disk, FDEL for deleting a file, FREN for renaming a file, SAVE, LOAD and APPEND for saving and loading programs on the disk and CHAIN for having one BASIC program call another. It also supports sequential disk files (which has caused confusion among some readers who thought, based on SSB advertising, that random files were also supported).

File commands include OPEN and CLOSE, READ and WRITE, RESTORE (for resetting a file

Listing 2.

```
                        NAM  SSB-TSC EDITOR
                        OPT  TAP,NOG
                    * ********************************
                    * * SMOKE SIGNAL  TSC/EDITOR    *
                    * ********************************
                    * ***** SSB-DFM FUNCTION CODES *****
0001                QS04W  EQU  1              OPEN FOR WRITE
0002                QSWRIT EQU  2              WRITE DATA
0003                QSWC   EQU  3              CLOSE FILE FOR WRITING
0004                QS04R  EQU  4              OPEN FOR READ
0005                QSREAD EQU  5              READ DATA
0006                QSRC   EQU  6              CLOSE FILE FOR READING
                    * ***** FCB LAYOUT *****
0000                XFC    EQU  0              FCB+0  FUNCTION CODE
0001                XES    EQU  1              FCB+1  ERROR STATUS RETURNED TO USER
0002                XUN    EQU  2              FCB+2  UNIT NUMBER
0003                XFN    EQU  3              FCB+3  FILE NAME
0009                XEN    EQU  9              FCB+9  EXTENSION NAME
000C                XFT    EQU  12             FCB+12 FILE TYPE
                    * ***** SSB-DFM ENTRY POINTS *****
7780                ODFM   EQU  $7780          INIT OF DFM
7783                CDFM   EQU  $7783          SSB-DOS CLOSE ALL FILES
7786                DFM    EQU  $7786          SSB-DISC FILE MANAGER
7283                ZWARMS EQU  $7283          SSB-DOS WARM START
72B5                ZLINEI EQU  $72B5          SSB-INPUT STRING INTO LINE BUFFER
7291                ZFLSPC EQU  $7291          LOAD FCB FROM LINE BUFFER
72A9                ZTYPDE EQU  $72A9          TYPE OUT ERROR MESSAGE
                    * ***** MIKBUG ENTRY POINTS *****
E07E                PDATA1 EQU  $E07E          MIKBUG PRINT STRING ROUTINE
                    * ***** OPEN FILE FOR READ *****
1491                        ORG  $1491
1491 CE 15 14       OR      LDX  #FCB          POINT TO OUR FCB
1494 86 04                  LDA  A #QS04R        CODE FOR OPEN SEQUENTIAL READ
1496 A7 00                  STA  A XFC,X         STORE FUNCTION CODE INTO FCB
1498 BD 77 86               JSR  DFM           CALL SSB-DFM
149B 26 52                  BNE  ERR           BRANCH IF OPEN ERROR
149D 86 05                  LDA  A #QSREAD       CODE FOR READ SEQUENTIAL
149F A7 00                  STA  A XFC,X         STORE FUNCTION CODE INTO FCB
14A1 39                     RTS
                    * ***** OPEN FILE FOR WRITE *****
14A2 CE 15 14       OW      LDX  #FCB          POINT TO OUR FCB
14A5 86 01                  LDA  A #QS04W        CODE FOR OPEN SEQUENTIAL WRITE
14A7 A7 00                  STA  A XFC,X         STORE FUNCTION CODE INTO FCB
14A9 BD 77 86               JSR  DFM           CALL SSB-DFM
14AC 26 41                  BNE  ERR           BRANCH IF OPEN ERROR
14AE 86 02                  LDA  A #QSWRIT       CODE FOR WRITE SEQUENTIAL
14B0 A7 00                  STA  A XFC,X         STORE FUNCTION CODE INTO FCB
14B2 39                     RTS
                    * ***** ASK TERMINAL FOR FILE NAME *****
14B3 CE 15 BA       ASK     LDX  #ASKMSG       POINT TO "NAME?" MESSAGE
14B6 BD E0 7E               JSR  PDATA1        MIKBUG OUTPUT STRING ROUTINE
14B9 BD 72 B5               JSR  ZLINEI        INPUT STRING INTO SSB LINE BUFFER
14BC CE 15 14               LDX  #FCB          POINT TO OUR FCB
14BF BD 72 91               JSR  ZFLSPC        LOAD FCB WITH INFO FROM LINE BUFFER
14C2 25 0E                  BCS  SERR          SYNTAX ERROR WAS FOUND
14C4 CE 15 14               LDX  #FCB          POINT TO OUR FCB
14C7 86 54                  LDA  A #'T           ADD 'TXT' EXTENSION
14C9 A7 09                  STA  A XEN,X         -
14CB A7 0B                  STA  A XEN+2,X       -
14CD 86 58                  LDA  A #'X           -
14CF A7 0A                  STA  A XEN+1,X       -
14D1 39                     RTS
14D2 CE 15 CA       SERR    LDX  #SYNTAX       POINT TO ERR MSG
14D5 BD E0 7E               JSR  PDATA1        MIKBUG OUTPUT STRING ROUTINE
14D8 20 D9                  BRA  ASK           GO ASK GUY AGAIN
                    * ***** READ/WRITE SUBROUTINE *****
14DA FF 15 DD       RW      STX  XSAVE         SAVE X-REG
14DD F7 15 DC               STA  B BSAVE         SAVE B-ACCUM
14E0 CE 15 14               LDX  #FCB          POINT TO OUR FCB
14E3 BD 77 86               JSR  DFM           CALL SSB-DFM TO READ/WRITE
14E6 26 07                  BNE  ERR           IF ERROR, GO REPORT IT
14E8 FE 15 DD               LDX  XSAVE         RESTORE X-REG
14EB F6 15 DC               LDA  B BSAVE         RESTORE B-ACCUM
```

back to the beginning), SCRATCH (for erasing a data file) and STATUS (for testing if any errors occurred during a file access).

SSB offers a number of other programs for their disk system. This includes the SE-1 Super Editor ($29), the SA-1 Super Assembler ($29), TP-1 Text Processor ($40), TD-1-D Trace-Disassembler ($26) and the SG-1-D Source Generator ($31). They also have a full FORTRAN compiler ($99).

Also available are a user's library disk ($20) containing additional transient commands and an SD-2 BASIC compiler ($325). The text editor, processor and assembler are written by TSC and, except for the way they use the disk and are loaded and started, are identical to the software available for the SWTP MF-68 disk. The BASIC compiler is written by Software Dynamics.

Other software is available from Ed Smith's Software Works; some SSB software comes from Ed Smith's.

Next to SSB, the most prolific supplier of DOS-68 compatible software appears to be Computerware Software Services. Their Random-Access Disk File BASIC ($90) supports not only sequential files, but also indexed sequential or true random-access disk files, where file entries can be changed without rewriting the entire file. This powerful BASIC interpreter is faster than standard SWTP BASICs. (There is a good chance that this BASIC will be offered by SSB as well during 1979.)

Computerware also offers a Disk Check File Maintenance system ($50), a Disk Mailing List system, several disks with transient commands, a BASIC renumberer program called RENBAS and various other disk-based programs.

Microware also offers SSB versions of the same software they offer for mini FLEX.

```
14EE 39              RTS
                * ***** COMMON ERROR REPORTER *****
14EF BD 72 A9 ERR    JSR  ZTYPDE        PRINT DISC ERROR CODE
14F2 BD 77 B3        JSR  CDFM          CLOSE ALL OPEN FILES
14F5 7E 04 41        JMP  EDIT          GO BACK TO MAIN PROGRAM
                * ***** INPUT 'EOF' ROUTINE *****
1458            READ25 EQU  $1458       RETURN TO PROCESS READ
1464            READ4  EQU  $1464       READ EOF
14F8 81 1A      EOF    CMP  A #$1A        TEST FOR 'EOF' CHAR
14FA 27 03           BEQ  X1            EOF HAS BEEN FOUND
14FC 7E 14 58        JMP  READ25        IF NOT, RETURN
14FF FF 15 DD X1     STX  XSAVE         SAVE X-REG BUFFER POINTER
1502 CE 15 14        LDX  #FCB          POINT TO OUR FCB
1505 86 06           LDA  A #QSRC         CODE FOR CLOSE READ SEQUENTIAL
1507 A7 00           STA  A XFC,X         STORE ITFUNCTION CODE INTO FCB
1509 BD 77 B6        JSR  DFM           CALL SSB-DFM
150C 26 E1           BNE  ERR
150E FE 15 DD        LDX  XSAVE         RELOAD X-REG
1511 7E 14 64        JMP  READ4         CONTINUE WITH 'EOF' PROCESSING
                * ***** CONSTANTS AND TERMINAL MESSAGES *****
1514            FCB    RMB  166          166 BYTE BLOCK FOR OUR FCB
0441            EDIT   EQU  $0441        PRINT '?' AND EDIT PROMPT
15BA 0D         ASKMSG FCB  $D,$A,0,0,0,0,0
15C1 46              FCC  'FILE:  '
15C9 04              FCB  4
15CA 0D         SYNTAX FCB  $D,$A,0,0,0,0,0
15D1 53              FCC  'SYNTAX ERR'
15DB 04              FCB  4
15DC            BSAVE  RMB  1            SAVE B-ACCUM
15DD            XSAVE  RMB  2            SAVE X-REG
15DF 0D              FCB  $0D            SET BEGINNING OF BUFFER
15E0            BUFFER EQU  *            EDIT BUFFER MARKER
                * ***** OVERLAY OLD CODE, AND ADD PATCHES *****
020C                 ORG  $020C          I/O VECTORS
020C 7E 14 DA        JMP  RW             READ ROUTINE
020F 7E 14 DA        JMP  RW             WRITE ROUTINE
0212 4F FF           FDB  $4FFF          ****** SET MEMORY TO BE USED
                *
0272                 ORG  $0272          COMMAND TABLE
0272 04 41           FDB  $0441          GAP IS NOW INVALID
                *
0358                 ORG  $0358
0358 CE 15 E0        LDX  #BUFFER        POINT TO BEGINNING OF USABLE MEMORY
                *
098B                 ORG  $098B          EXIT COMMAND
098B 7E 72 B3        JMP  ZWARMS         GO BACK TO SSB DOS
                *
13D3                 ORG  $13D3          WRITE COMMAND
13D3 BD 14 B3        JSR  ASK            ASK FOR OUTPUT FILE NAME
13D6 BD 14 A2        JSR  OW             OPEN FILE FOR OUTPUT
13D9 01              NOP
                *
13FA                 ORG  $13FA          WRITE EOF
13FA CE 15 14        LDX  #FCB           POINT TO OUR FCB
13FD 86 03           LDA  A #QSWC          CODE FOR CLOSE WRITE SEQUENTIAL
13FF A7 00           STA  A XFC,X          STORE FUNCTION CODE INTO FCB
1401 BD 77 B6        JSR  DFM            CALL SSB-DFM
1404 27 03           BEQ  X2             IF OK, RETURN
1406 7E 14 EF        JMP  ERR            IF NOT, REPORT ERROR
1409 39         X2   RTS
                *
142D                 ORG  $142D          READ COMMAND
142D BD 14 B3        JSR  ASK            ASK FOR INPUT FILE NAME
1430 BD 14 91        JSR  OR             OPEN FILE FOR INPUT
1433 01              NOP
                *
1454                 ORG  $1454          READ EOF
1454 7E 14 F8        JMP  EOF            GO TEST FOR EOF
                *
146F                 ORG  $146F          DEVICE CONTROL CHARACTERS
146F 01              NOP
1470 01              NOP  NOP OUTPUT OF TURN-OFF CHAR
1471 01              NOP
                *
                     END
```

Microware also offers SSB versions of the same software they offer for mini FLEX.

Advanced DOS-68 Programming Example

Most users will probably use the supplied software and program mostly in BASIC, but the system can also be used in assembly-language programs. The DOS-68 manual has a completely different orientation than SWTP/TSC's FLEX manual. Whereas the FLEX manual is written for the casual user and is supplemented by an advanced programmer's manual, the SSB DOS-68 manual is much more detailed and also lists the information needed by advanced programmers.

The diskette is maintained by a disk-file-management set of routines. They are accessed via a jump table that supplies entry points for various file-handling functions. These include opening and closing files, reading, writing and several other special I/O requests.

Like the SWTP disk, all file functions are channeled through a user-created work area called File Control Block (FCB). An FCB is required for each active file the user is accessing. This area acts as an interface to DFM to communicate the user's request for I/O. The DFM routines, in turn, use it as a buffer to hold data, letting the user access a character at a time. Several routines are available in the DOS Monitor to assist the user in correctly loading the FCB with the needed information.

The standard DOS supports sequential processing, and the expanded DOS adds random accessing as well. Space compression may be requested for certain text files, along with other special processing techniques.

An example of using the Smoke Signal DFM routines is shown in Listing 2. This listing shows the patches to the cassette version of the TSC Text Editor to run on an SSB disk system.

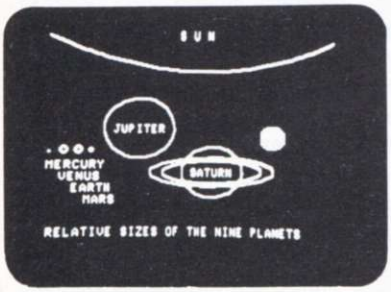
The beginning of the listing shows the entry points into the DFM routines, along with those of MIKBUG that are used. In addition, some offsets into the FCB are defined for later usage. The next segments of software reflect the disk routines to open a file, close a file, prompt the user for a file name and perform other common I/O. These newly added segments are referenced by some patches to the editor code. At the end of the listing are the patches that reset the pointers within the editor to reflect the new disk software.

This listing is similar to that of the FLEX patches. There are a few differences, mainly in the FCB offsets and in entry point addresses. The ZLINEI routine is used to accept data from the terminal and store it into the line buffer. The monitor routine ZFLSPC is used to parse through the buffer and load the FCB with the proper information. The extension must be loaded by hand, as there is no routine to perform this function. In addition, the files are stored as memory image binary, rather than compressed ASCII.

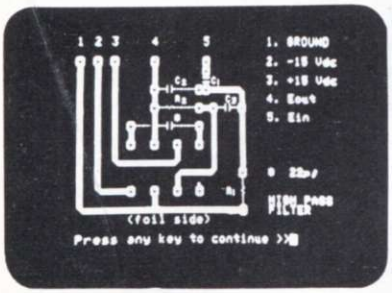
As you can see, though the DOS-68 and FLEX systems are different, they are set up in similar ways. Thus, any software created for one system should be fairly easily modified for the other.

Next month, we will examine the features of the Percom LFD-400 disk system and conclude this two-part article by weighing the advantages and disadvantages of each of the three systems. ■

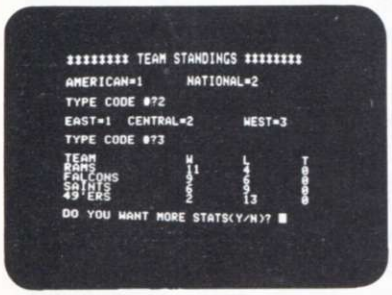
Software for the APPLE II



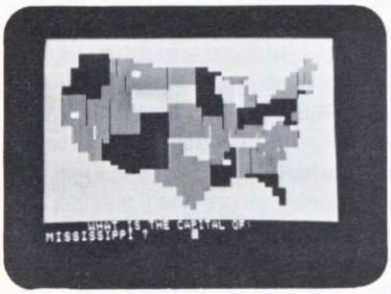
THE PLANETS $15.95



ACTIVE FILTERS $24.95



FOOTBALL PREDICTIONS $19.95



STATE CAPITALS $9.95

FORMAT

PROGRAMMA's FORMAT (Version 1.0) is a command oriented text processor designed to be fully compatible with PIE (PROGRAMMA Improved Editor).

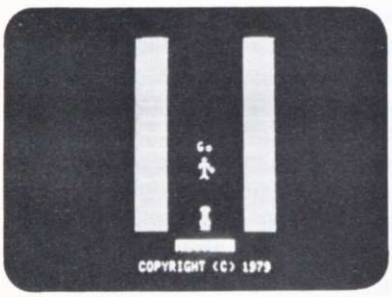
FORMAT's system of imbedded commands (within the text) give it an ease of operation similar to text formatters found on some mini-computers.

FORMAT features right margin justification, centering, page numbering, anda auto-paragraph indent.

The following commands are available with FORMAT:

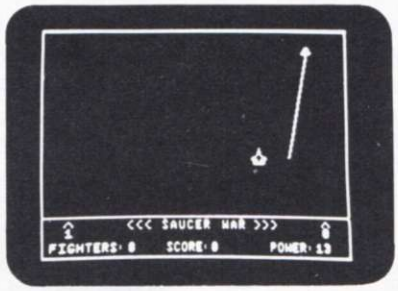
.ad		Begin adjusting right margins
.bp	n	Begin page numbered n
.br		Cause a line break
.ce	n	Center next n lines without fill
.fi		Start filling output lines
.fo	t	Foot title becomes t
.he	t	Head title becomes t
.in	n	Indent n spaces from left margin
.li	n	Literal, next n lines are text
.ll	n	Line length including indent is n
.ls	n	Set line spacing to n
.ml	n	Top spacing including head title
.m2	n	Spacing after heading title
.m3	n	Spacing before foot title
.m4	n	Bottom spacing including foot title
.na		Stop adjusting right margins
.nf		Stop filling output lines
.pl	n	Page length is n lines
.pp	n	Begin paragraph= .sp, .fi, .ti n
.sp	n	Space down n lines, except at top
.ti	n	Temporary indent of n
.ul	n	Underline next n input lines

All orders include 3% postage and handling.
Apple II is a registered trademark of Apple Computer, Inc.
California residents add 6% Sales Tax
VISA & MASTERCHARGE accepted.

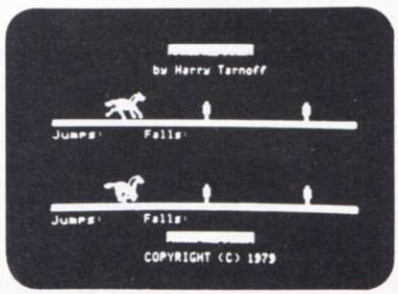


SPEEDWAY $15.95

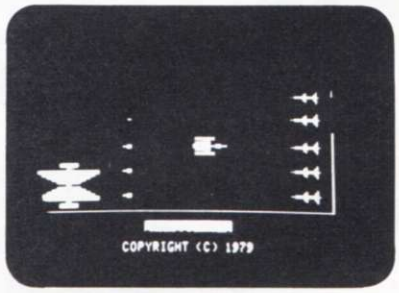
JOY STICK $49.95
EXPAND-A PORT $49.95



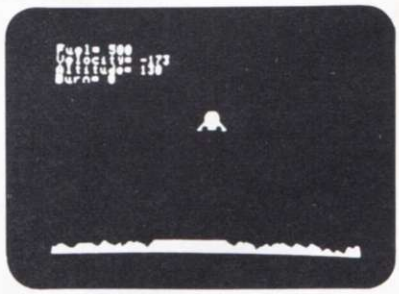
SAUCER WAR $15.95



CANTER DOWNS $15.95



BATTLESTAR I $15.95

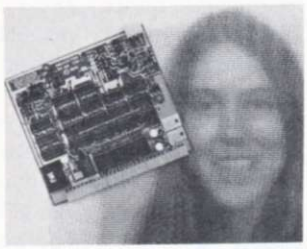


LUNAR LANDER $9.95

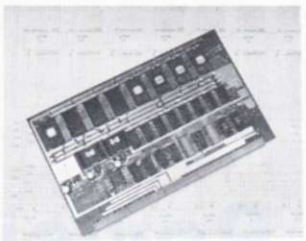
PROGRAMMA INTERNATIONAL, Inc.
3400 Wilshire Blvd.
Los Angeles, CA 90010
(213) 384-0579
384-1116
384-1117 ↙P48

Dealer Inquiries Invited

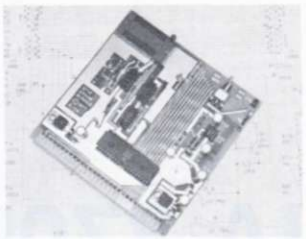
PROGRAMMA Software Products



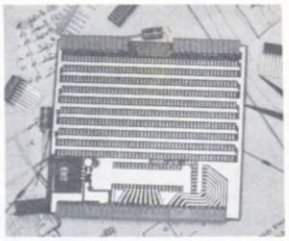
DS-68 DIGISECTOR



PSB-08 PROM SYSTEM BOARD



B-08 EPROM PROGRAMMER



UIO UNIVERSAL I/O BOARD

INNOVATIVE PRODUCTS FOR 6800 USERS

DS-68 DIGISECTOR is a random access video digitizer featuring 256 × 256 picture element scan and 64 levels of grey scale, with conversion times as low as 3 microseconds per pixel. It accepts either interlaced (NTSC) or non-interlaced (industrial) video input. Use it for computer portraiture, moving target indicators, precision security systems, fast to slow scan conversion. . . with clever software, the Digisector can read just about anything. Truly a professional tool at a price you can afford. $169.95

DM-85 DISK MIXER is an add-on board for the Smoke Signal Broadcasting BFD-68A Disk Controller which allows operation of both 8" and 5" drives. Controller mode (8" or 5") is selected on a drive-by-drive basis, so any mix of 5" and 8" drives is allowable. The 2" × 3" PC board mounts inconspicously on the back of the BFD-68A. Its operation is completely transparent to software. An oscilloscope is required for the setup procedure. Kit Price: $39.95

B-08 2708 EPROM PROGRAMMER is a compact unit that fits in the 6800's I/O slot. A safety switch and LED indicator provide control over the high programming voltage generated on board. An industrial quality Textool socket and extended board height allow effortless PROM insertion and retrieval. Fully commented source listings of U2708 is included in the Owner's Manual. $99.95

U2708 utility for testing, burning, verifying and copying 2708s in EPROM. $29.95

M6809 EMULATOR is a machine language program that will emulate all of the functions of the Motorola 6809 third generation microprocessor. Developed for use on any 6800 system, the program allows software development and debugging. The 3K byte program is complete with a 6809 mini-monitor and single-step trace routines. Fully commented source listing included. Specify Smoke Signal Broadcasting or FLEX™ disk, or KCS cassette. $49.95

PSB-08 PROM SYSTEM BOARD features 1K of high speed, low-power RAM and space for up to 8 2708 EPROMs, both DIP-switch addressable to start on any 8K boundary in memory. The exclusive I/O select feature allows you to move I/O locations up to any unused 1K block in the EPROM memory space. This permits memory expansion to a full 56K of contiguous user RAM. $119.95

UIO UNIVERSAL I/O BOARD helps you with your custom interfaces. It has space for a 40-pin wire wrap socket into which you may plug any of Motorola's 40 or 24-pin interface chips. All data and control lines are connected to the appropriate edge connector pins. All other bus connections are brought out to a 16-pin socket pad. +5 volt regulator and all Molex connectors are provided; regulated +5 and ground are bused among the locations for up to 35 14-pin ICs. $24.95

THE MICRO WORKS

✓M31

P.O. BOX 1110, DEL MAR, CA 92014 714-756-2687

PAIA 8700 Revisited

Back in August 1979 (p. 94), Rod presented an overview of an under-200 computer, the PAIA 8700. This time, he's going to cover a specific application for the 8700: making music.

Rod Hallen
Road Runner Ranch
PO Box 73
Tombstone AZ 85638

In a recent article describing the PAIA 8700 computer, August 1979, p. 94, I touched on its physical characteristics, operating techniques and some possible uses. In this article I'll explore one particular application.

Computer music seems to be a popular subject. I received more phone calls and letters as the result of my computer music article ("Music, Music and More Music," *Kilobaud*, November 1978, p. 82) than anything else I've ever written. Therefore, the first thing I tried to do with the 8700 was make some music.

Generating Computer Music

There are two general methods of generating music with a computer. First, the processor can switch one of its external leads on and off at an audio rate. When amplified, this is sound; if done right, it is also music. The second way is to use the computer to control an electronic organ or a synthesizer of some kind.

The late, great "Music System" from Software Technology is an example of the first method. It has the advantage that it is easy to use and that no external hardware other than an amplifier is required. In fact, when not much volume was needed, I drove a small speaker directly with TTL logic. The disadvantage of this method is the limited control of the type of sound produced. On the other hand, using the computer to control an organ or a synthesizer allows you much more freedom of expression. After some experimentation I decided on this second method.

Since I didn't have a synthesizer, I first had to build one. I ordered the Top Octave Experimenter's kit from PAIA to use as a starting point (see Table 1). I also sent for the cassette interface option kit and enough RAM to fill up the board. The 8700 comes with 512 bytes of RAM with space on the board for another 512 bytes for a total of 1024. This RAM is addressed from 0 to 03FF hex.

I quickly installed the cassette kit, which worked the first time. A bonus of the cassette option is a solid-state sounder that beeps whenever a key is touched. It also allows you to hear the recording and playback of your programs. Doubling RAM to 1K required installing four IC sockets and four 2112s ICs.

The top octave kit contains a small PC board, a 4001 CMOS IC and a Mostek 50240 top octave generator. The latter is actually a divider that puts out 13 tones derived from a common clock frequency. If one of the 13 tones is on frequency then all of them have to be right. Using a 2.00024 MHz clock, you can produce the top keyboard octave.

Fig. 1 is a pin-out drawing and block diagram of the 50240. You can change the clock fre-

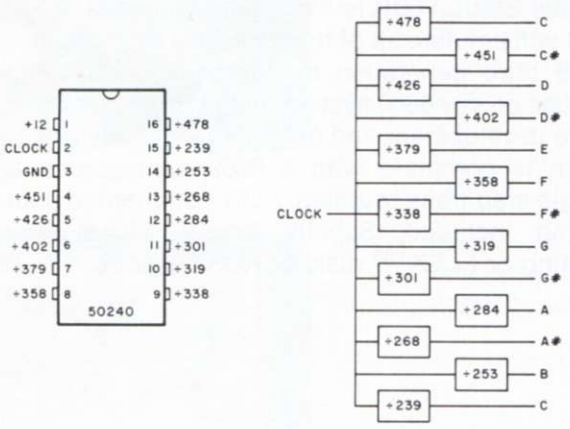


Fig. 1. A pin-out drawing and a block diagram of the Mostek 50240 top octave generator. Inputs are + 12 volts, ground and a clock. Outputs are 13 equally tempered semi-notes making up one octave plus one note.

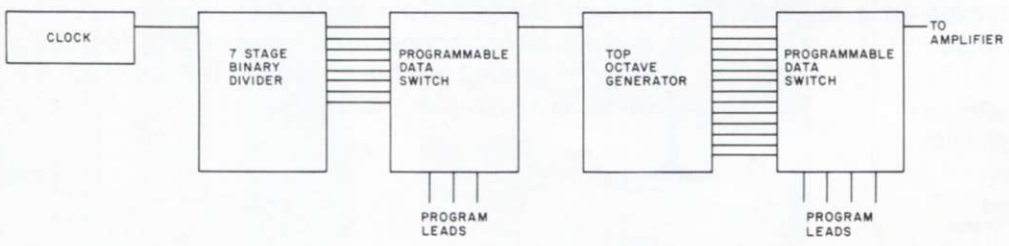


Fig. 2. The block diagram of an effective music generator. The program leads could go to a switch console or a piano-type keyboard, but the real intent is to interface them with a computer output port.

quency to suit your requirements. With a 2.00024 MHz clock and a string of binary dividers any note on a piano or organ keyboard can be produced. Running these notes into various synthesizer modules will allow you to come up with any sound desired.

Fig. 2 is a block diagram showing how a piano keyboard could be reproduced. This simple but sophisticated system is described in "Try Computer Composition" by Kenny Winograd (July 1977 *Kilobaud*, p. 102).

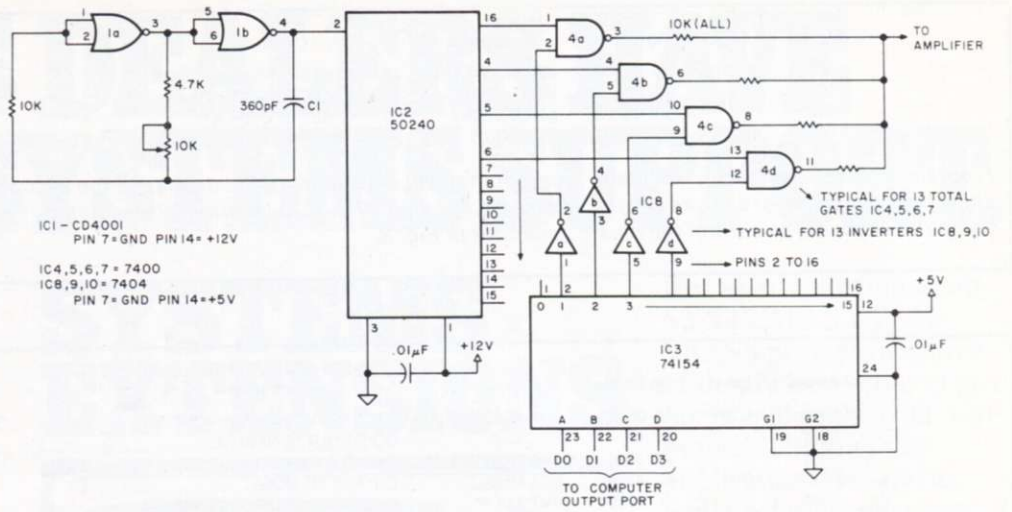


Fig. 3. Schematic of a simple computer-controlled music generator. The TTL ICs could be replaced with CMOS.

Hardware

For experimentation purposes I put together the simple circuit of Fig. 3. The 4001 and the 50240 go on the PC board that comes with the Top Octave Experimenter's kit, but the rest of the ICs must be mounted in some other fashion. In my case they went onto my IC breadboard.

Two of the gates of the 4001 (IC1) are used as an oscillator with the 10k pot for frequency adjustment. A frequency of 224,960 Hz fed from the oscillator to the clock input of the top octave generator (IC2) will result in the middle octave of a piano with A at 440 Hz. Changing the value of C1 will move the range of the oscillator. A smaller value will increase the frequency and vice versa.

With C1 equal to 360 pf (two 180 pf disk capacitors in parallel), the 13 semitones of the middle keyboard octave are present at the output leads of the 50240. It is just a question of feeding the desired tone to the amplifier. The various 7400 gates (ICs 4, 5 and 6) are used to switch one tone at a time to the common output point. The actual selection is determined by the condition of the four data leads (D0, D1, D2 and D3) from the computer output port.

The 74154 (IC3) is a four-line-to-16-line decoder. If the binary code on lines A, B, C and D is zero (see Table 2), then all output lines will be high except 0. If the binary code is 1, then all lines except 1 will be high. Thus it is possible to control up to 16 notes.

An inverter (ICs 8, 9 and 10) is inverted in each output line from the 74154 to satisfy the requirements of the 7400 gates. Since the signal we want to pass is positive-going, it will only pass through the gate when the control lead is also high (see Table 3). The output of the 7400 will be inverted, but this does not present a problem.

I used a combination of TTL and CMOS IC logic because that is what I had available. You could replace the TTL ICs with their CMOS equivalents. This would not only cut way down on the current requirements but

Leads	Output	Lead	Value	
D C B A	LOW		H	L
L L L L	0	A	1	0
L L L H	1	B	2	0
L L H L	2	C	4	0
L L H H	3	D	8	0
L H L L	4			
L H L H	5			
L H H L	6			
L H H H	7			
H L L L	8			
H L L H	9			
H L H L	10			
H L H H	11			
H H L L	12			
H H L H	13			
H H H L	14			
H H H H	15			

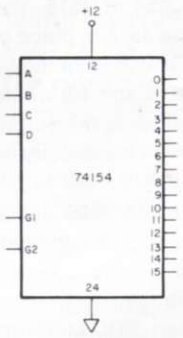


Table 2. Truth table for the 74154 four-line-to-16-line decoder. G1 and G2 must be held low at all times or all output leads will stay high no matter what the condition of A, B, C or D is. Only one output lead can be low at any one time. Leads A, B, C and D are coded in binary coded decimal (BCD). The leads have the value shown on the right. The values of all of the leads in an H (high) condition are added together to get the equivalent decimal value. For instance, if A and C are high then the decimal value is 5.

PAIA 8700 Computer/Controller Kit		$149.95
Cassette Interface Option Kit		22.50
Power Supply Kit		24.95
Top Octave Experimenter's Kit		14.95

Write to PAIA Electronics, Inc., PO Box 14359, Oklahoma City OK 73114, for their complete catalog of computer and music-synthesizer products.

Table 1.

Lead		Output	NAND Gate
A	B	C	
L	L	H	
L	H	H	
H	L	H	
H	H	L	

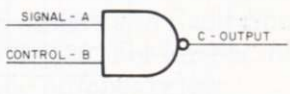


Table 3. If the control lead (B) is held low, the output will never change no matter what the signal lead (A) does. If the control lead B is held high, then the output will change as the signal does, but it will always be inverted by the 7400 NAND gate.

Address	Label	Op Codes	Mnemonics	Comments
0020	LOOP	20 00 FF	JSR KEYBOARD	CALL KEYBOARD SUBROUTINE
0023		B0 FB	BCC LOOP	LOOP IS NOT NEW KEY
0025		8D 20 A8	STA DISPLAY	STORE KEY IN DISPLAY
0028		4C 20 00	JMP LOOP	GET ANOTHER KEY

Program A. A 6502 assembly-language program that will cause the digital display of the 8700 to show the key number of each key as it is pressed. Changing 20 to 40 at location 0026 will allow the computer keyboard to control the music generator in Fig. 3.

Address	Label	Op Codes	Mnemonics	Comments
0020	LOOP	20 00 FF	JSR KEYBOARD	CALL KEYBOARD SUBROUTINE
0023		B0 02	BCC OUT	GO TO OUT IF NEW KEY
0025		A9 00	LDA NOTONE	CLEAR A REGISTER
0027	OUT	8D 40 A8	STA MUSIC	OUTPUT TO MUSIC
002A		4C 20 00	JMP LOOP	CHECK KEYBOARD AGAIN

Program B. A slight modification of Program A disables the tone when a key is not being pressed by putting out 0 (no tone) when the keyboard subroutine returns with the carry set (no new key).

would also eliminate the need for the +5 volt supply since the whole thing could run on +12 volts.

The 7400s could be replaced by 74C00s or 4011s, the 7404s by 74C04s or 4069s, and the 74154 by 74C154 or 4515. You could also use 4514 in place of the 74154, in which case the inverters (ICs 8, 9 and 10) would not be needed since the output leads of the 4514 are already inverted.

The 74C00 and 4000 series ICs are not pin-for-pin compatible with their 7400 counterparts and are generally more expensive than TTL. Motorola identifies their CMOS products with the prefix MC1. Thus the 4011 would be MC14011. Other manufacturers prefix the device number with CD—thus CD4011.

Without connecting Fig. 3 to the computer, you can test the circuit by grounding one or more of the data leads. In fact, for a simple piano you could remove the 74154 and connect its output leads directly to a group of switches (Fig. 4), eliminating the computer completely. However, if this is possible why do we even need the computer?

Yes, you can play this like a piano with the switches in Fig. 4, but if you are not a musician, the result may be unpleasant to listen to. By storing the notes of a composition in memory, you can let the computer keep track of the timing and play them for you. The tempo can easily be changed, and you can save the music code on tape in order to play it again another time.

The Winograd article mentioned previously does a good job of covering both hardware and software, so I won't try to go into too much detail here. However, let's see how the 8700 can be used to control the top octave generator.

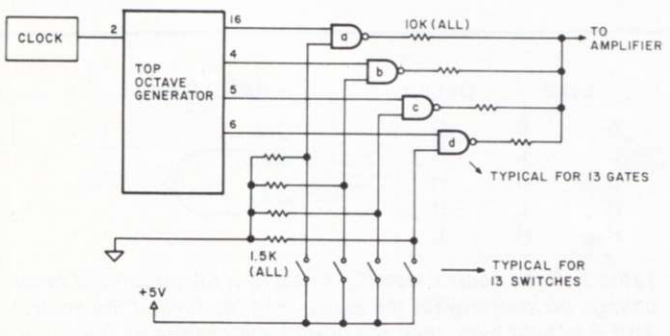


Fig. 4. A 13-note keyboard. More than one note can be played at a time in this configuration.

Software

The 8700 Piebug monitor (in ROM) has a subroutine at FF00 that scans the keyboard, and when it finds a key depressed, it returns with the number of the key in the A register. It also clears the carry flag if this is a new key since the last scan.

Program A is a simple 6502 assembly-language program that reads the computer keyboard and sends what it sees to the digital display. It does this by calling the keyboard subroutine, checking to see if it is a new key, and if it is, it stores the key number at A820 hex, which is the address of the digital display. As you press each key, its number appears on the display.

The parallel output port address is A840 hex, and if we change the 20 hex to 40 hex at location 0026 in Program A, then the key number will be sent to the output port instead of to the digital display. With our little music board connected to the lowest four bits of the output port, we can control the notes from the computer keyboard.

One disadvantage of Program A is that a note will continue to sound until a new key is pressed because the 7800 output port is a latched port that will hold its present value until it receives a new value. Program B takes care of that by shutting off the tone, except while a key is actually being pressed. It does this by sending a 0 to the output port between notes.

A 0 from the keyboard will shut off the music since none of the gates are connected to the 0 lead of the 74154. Keys 1 through D will play the 13 notes of our scale (see Table 4). This is exactly like a piano keyboard, but as I've mentioned it is better and more fun to teach the computer to play for us. This is accomplished by storing notes in memory and then feeding them to the music board along with the proper timing for each note.

Using the techniques presented in Winograd's article, you can put together an inexpensive but sophisticated music-generating system. The 8080 software in that article would have to be rewritten in 6502 assembly language, but it is a simple program and shouldn't present any problems.

Conclusion

These same methods could also be used to control other devices. In fact, anything that can be turned on or off or adjusted electrically is a candidate for computer control, which is what the PAIA 8700 was designed for.

Computer music is an enjoyable subject for software and hardware experimentation. It also satisfies the urge to do something creative with your personal computer. When friends ask if your computer can do anything besides play games, it is great fun to pop in a music cassette and let it play a tune for them.■

Key	Note	Key	Note
0	NONE	7	F#
1	C	8	G
2	C#	9	G#
3	D	A	A
4	D#	B	A#
5	E	C	B
6	F	D	C

Table 4. List of the notes that correspond with the first 14 keys on the 8700 keyboard.

TBS DEALERS HAVE FULL SYSTEMS SUPPORT.

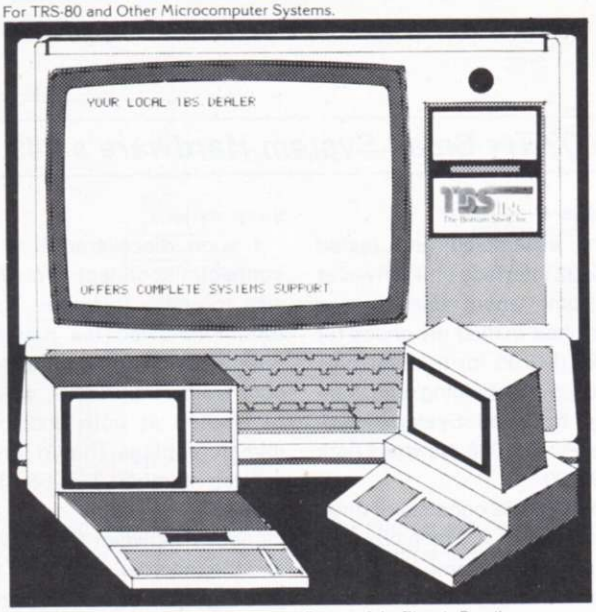
You've just purchased a Centronics printer. Where do you get the printer supplies needed to keep it functional? You've just purchased a disk drive. Where do you find a source of quality diskettes and storage devices to keep them organized? You have invested over $1,000 in your system. Where will you find all the necessary items to keep it operational? The answer: TBS and its affiliate DEALERS.

The LIBRARY 100 Software Pack was published in November of 1978. Since that time, it has been widely accepted throughout the United States and 31 countries as a major advancement in microcomputer software. Now, we at TBS have expanded our endeavor to better meet the product needs of our customers.

SYSTEMS EXTENSIONS

$3.00

For TRS-80 and Other Microcomputer Systems.



Computers of the Past
Computers of the Present
Computers of the Future
Methods to Program Your System
Software Background Report
Preparation for Programming
Standards for Professional Programming
Security of Business Data and Programming
Purchase, Care and Maintenance of the Business Computer
Review of the Electric Pencil
The Diskette Revolution
The TRS-80 and the Business Community
Computer Aided Instruction
The Microcomputer and the Wall Socket
Level II Index
Random Ramblings

A camera is useless without film; a typewriter is useless without paper; a computer system is useless without the necessary support to make it completely operational. TBS is leading the way in systems support, and we are proud to announce a new concept for the microcomputer industry - **FULL SYSTEMS SUPPORT**. Mainframe and minicomputer industries have been following this concept for years. However, until now it has not been available for microcomputer systems. TBS Full Systems Support is aimed at helping you, the computer owner, acquire the necessary support items quickly and inexpensively through a network of affiliate dealers nationwide.

We also realize that there is limited material available on computers in general. To fill these needs we have produced **SYSTEMS EXTENSIONS**. This is a unique publication which provides a theoretical computer background for the novice computer user and some specific points of programming interest for TRS-80 users. Seventeen articles are included in 63 pages of solid text.

But even more important, **SYSTEMS EXTENSIONS** provides a source of supply for a series of products applicable to all computers. Through this publication TBS and its affiliate dealers offer over 300 items including software, diskettes, cassettes, printer supplies, storage units and furniture. In other words, TBS and its dealers stock everything to make the microcomputer a full scale business system.

At the heart of your computer system is the software. We now offer the following programs for the TRS-80: **TBS Business Mail System** for dual disk and printer; **Check Register Accounting System** for dual disk and printer; **Basic Toolkit**, a programmers aid; **System Doctor**, a computer diagnostic program; **Checkbook II**, cassette based personal finance; **Information System**, an "in-mem" data base manager; **Analysis Pad**; a columnar calculator; **Terminal Control**, for RS-232 tele-communications; **Exerciser**, for establishing physical fitness goals and regimens; plus, **Electric Pencil**; **Fortran** by Microsoft; **NEWDOS+**; and of course, the **Library 100**. Details in full are included in **SYSTEMS EXTENSIONS**.

We at TBS believe in the microcomputer revolution. Our purpose is to serve you, the microcomputer owner and user, by supplying the necessary items you need. To this end we have published **SYSTEMS EXTENSIONS**, and it is available nationwide through TBS dealers and Radio Shack Associate stores for only $3.00. For further information please call us at the number below.

TBS INC.

The Bottom Shelf, Inc. ✓B33
P.O. Box 49104
Atlanta, Georgia 30359
Phone: 404-939-6031

✓ Reader Service—see page 227

An Inexpensive TRS-80 Printer Interface

Why spend $300? Try Small System Hardware's $49.95 alternative.

Sherman P. Wantz
424 NW Lakeview Drive
Sebring FL 33870

If you own a "bare-bones" TRS-80 microcomputer system, you will eventually want to buy Radio Shack's $300 expansion interface to provide you with additional memory space, a floppy disk capability, an RS-232 serial port and a parallel port to feed a fast line printer. But suppose you have a printer and would like to connect it to your TRS-80 but can't see your way clear at the moment to spend $300?

That oversimplified scenario described my condition before I read the ad in *Microcomputing* about Small System Hardware's TRS-232 printer interface unit. I learned that for a modest investment I could give my TRS-80 a hard-copy capability, thereby satisfying at least a part of the advantage I had hoped to achieve by buying Radio Shack's expansion interface.

In response to the inquiry I mailed, using *Microcomputing*'s Reader Service card, Small System Hardware sent me a brochure that described its TRS-232 in detail. The brochure convinced me that the TRS-232 was what I needed to connect my TRS-80 microcomputer to my ASR-33 teleprinter so that I could use my Level II BASIC's LLIST and LPRINT commands.

Hardware

The assembled and tested TRS-232 interface unit arrived at my home about three weeks after I had mailed my check for $51.95 ($49.95 for the unit plus $2 to cover shipping and handling) to Small System Hardware, PO Box 366, Newbury Park CA 91320.

The TRS-232 consists of three parts: a 1 × 2 × 3 inch box containing a 741 op amp IC, a transistor, five diodes, resistors and capacitors—all mounted on a small circuit board; a cassette tape that contains a program written in Level II BASIC; and a ten-page booklet that includes instructions, a schematic diagram of the unit, a program listing in BASIC and an assembly-language source listing.

As the instruction booklet explains, the TRS-232 obtains its signal via the gray cord that normally connects the TRS-80 computer to the auxiliary jack of the tape recorder. To use the printer, you must remove the miniature plug covered by a gray sleeve from the recorder's auxiliary jack and plug it into either of the two jacks that are mounted at each end of the interface unit's case. The jacks are connected in parallel so it makes no difference which one you use.

If you should want to use the tape recorder to dump (CSAVE) a program in memory to tape, you must remove the miniature gray plug from the TRS-232 jack and reinsert it in the recorder's auxiliary jack.

I soon discovered that the connect/disconnect procedure was too easy to forget. Therefore, I followed the manufacturer's advice and built a patch cord from audio cable, which I terminated at both ends with miniature plugs (Radio Shack catalog number 274-286). I installed the patch cord permanently between the TRS-232 and the recorder's auxiliary jack (see Fig. 1) and can now print or record programs without bothering to connect or disconnect signal cords.

Power for the interface components is obtained through a line-cord connection that plugs into any 115 volt wall outlet. The TRS-232 contains no off-on switch. It is on when it is plugged into the wall outlet.

The line-cord plug houses a small transformer. Current drain through the transformer is so small that the housing barely feels warm to the touch.

The output signal produced by the interface unit is fed to the printer via a 25-pin BD-25 female connector that is mounted along one edge of the TRS-232's case.

Since the manufacturer assumes that your printer is equipped with a BD-25 male connector, he provides none. I bought the appropriate male BD-25 connector for $2 at the Orlando hamfest flea market. New connectors, some complete with cable, are advertised regularly by suppliers whose ads appear in this magazine.

Only two signal connections had to be made to the BD-25 plug to tie the interface unit to my teleprinter. Pin 9 provided −12 volts dc, and pin 10 provided +12 volts dc. These connections supplied the 20 mA loop current needed to operate the ASR-33 Teletype. Consult your teleprinter manual for appropriate connection points to be made on your machine's terminal board.

The pins of the BD-25 female socket mounted on the TRS-232 are not numbered. Fortunately, however, those pins on the matching male connector are numbered; this helped me to make the correct coupling between the interface unit and my teleprinter.

If your printer does not require 20 mA of driving current—as does my machine—your output signal connections to the interface unit will differ from those I have just described. Small System Hardware claims to have used their interface unit to drive RS-232 compatible printers of all types. RS-232 is the specification number that was given by the Electronic Industries Association to standardize voltage and impedance levels for transmission of digital data.

Please note that the TRS-232 interface unit provides a "receive only" one-way printer capability, so don't expect it to

permit you to use your printer's keyboard to send characters to your computer.

Software

Once you have your interface hardware installed between your TRS-80 and your printer, you will be ready to load (transfer from cassette tape to computer memory) the Level II BASIC program that is supplied with the TRS-232. This program converts parallel data into serial form to operate your printer.

When you apply power to your Level II BASIC microcomputer, the words "MEMORY SIZE?" appear on your video monitor's screen. When these words appear, you will be expected to respond appropriately to reserve memory space for the TRS-232 print function program.

If your system contains 4K bytes of memory, you should type "20360" to reserve space for the 110 bytes required by the program; if your machine contains 16K bytes of memory, you should type "32650." Press the Enter key. Now you are ready to CLOAD the 52-line TRS-232 Level II BASIC program from cassette tape into computer memory.

My program loaded without problem at a volume setting of "4" on my CTR-33 recorder. For those who use the CTR-41 recorder, the manufacturer recommends a volume setting of about "6." After the program has loaded, type "RUN" and press the Enter key.

In succession, the program will ask four questions. Your answers to three of the questions will configure the program to operate with your particular printer. Your response to the questions will: reconfirm the amount of memory space you reserved for the program when you turned your system "on"; specify the baud rate at which your printer operates; declare whether or not a line-feed signal must be sent to the printer following each carriage return; and specify the number of nulls your printer requires (if any) after each carriage return. Once you have answered these four questions, the BASIC program "pokes" your answers—using machine language—into reserved memory.

To test the operation of the printer interface hookup, turn your printer's motor on, type LLIST on your TRS-80 keyboard and press the Enter key. You should see the TRS-232 BASIC program that you just loaded into memory being transferred from computer memory to your printer.

Since the TRS-232 BASIC program has served its purpose after it has "poked" your printer's specifications into reserved memory space, it is no longer needed. Type NEW to erase it before you load the Level II BASIC program you intend using with your printer.

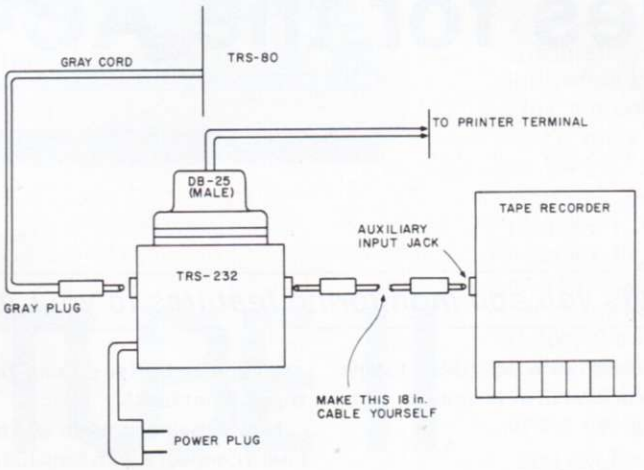


Fig. 1. TRS-80 connections that place the interface in series between the TRS-80 and the two output devices—printer and recorder.

Documentation

In addition to containing instructions for connecting the interface unit to your printer and for using it, the ten-page booklet that comes with the TRS-232 contains several noteworthy extras. If you are curious about how the TRS-232 works, a section of the instruction manual, entitled "Theory of Operation," will explain it to you in simple terms. In this section of the manual, the manufacturer points out that the TRS-232 is not certified as an interface that will meet all EIA Standard RS-232 requirements and specifications.

The schematic included with the manual shows how simple the design really is. The diagram promotes the idea—whether realistic or foolhardy—that anyone who can read a schematic should be able to troubleshoot and repair the interface unit. There should be no reason for anyone to return the Level II BASIC initialization program cassette to Small System Hardware because he can't load it into his computer.

Since the instruction manual contains a listing of the entire program (52 lines long—including 16 REMark lines that need not be used), it would be far simpler to type the program into memory using the computer's keyboard and then CSAVE it using your own computer, recorder and cassette tape. Because of slight record-head-alignment differences that exist among tape recorders, it is almost always easier to CLOAD tapes made on your own recorder than it is to load from tapes recorded on other machines.

Having the Level II BASIC listing in your instruction manual gives you peace of mind, too. Even if you do inadvertently store your initialization program cassette next to your TRS-80 system's power supply and partially erase it, you can always regain the use of your interface unit by typing and rerecording the program.

The instruction manual contains a special bonus for those who do their programming in Z-80 machine language. A complete listing in assembly language and machine code of the relocatable driver for the TRS-232 is provided. The code can be placed anywhere in memory since no calls and only relative jumps are used.

Baud rate selection (110 to 9600), line feed and nulls following carriage return options are provided, just as they are in the Level II BASIC program. Therefore, with the TRS-232 interface, you can obtain a print capability with your machine-language programs as well as with those that are written in Level II BASIC.

Conclusion

If you have a printer that you want to use with your TRS-80 system, you could hardly ask for a more reliable unit to tie computer and printer together than the TRS-232. Small System Hardware claims to have used the TRS-232 to interface with Diablo, Texas Instruments, Centronics, Teletype and Selectric printers.

Since the TRS-232 is not a kit—it comes fully assembled and tested—you will need no special skill, beyond the ability to follow written instructions, to install it in series with your TRS-80 microcomputer's output cable (the gray cord that terminates in a miniature plug enclosed in a gray plastic sleeve).

When you inspect the TRS-232 hardware housed in the 1 × 2 × 3 inch box, you may feel that the $49.95 price you paid for the unit is a bit steep. The price was $39.95 in December 1978 and jumped $10 at the beginning of 1979. However, when you remember that your purchase price also paid for a software package that will allow the same hardware unit to connect your TRS-80 to the next serial printer you acquire, the price doesn't seem too far out of line.

I am quite pleased with the hardware, software and instruction manual that comprise my TRS-232 interface. For my $50 interface investment, I obtained the use of a teleprinter that cost at least ten times that much. Particularly for a computer hobbyist, that's not a bad investment ratio. ■

Jerry Sorrels
6266 Banner Ct.
Riverside CA

Eyes for the AC-30

A simple modification lets you add monitoring features to your AC-30 cassette interface.

For over a year I had been using a Bit Boffer (see "Build a Bit Boffer," *Byte*, March 1976) to load programs into my SWTP 6800 computer. It worked just fine but lacked the ability to change from play to record and to start and stop the recorder under software control.

At first this was no problem . . . I had enough on my hands just getting the programs loaded and working. But as time went on, I became smarter and lazier.

Enter the SWTP AC-30 audio cassette interface. After using it for a short time, I missed one of the features of my old Bit Boffer: the ability to display the data on my SWTP CT-1024 terminal at the same time it was being loaded into the computer. This feature lets you verify if the tape is being read properly. Also, you know when the end is nearing.

But its best feature is that it gives you something to look at while waiting for the program to load. 300 baud is just a little slow!

After looking over the SWTP 19 × 25 inch schematic (nice and big), I came up with a simple way to add this monitoring feature to the AC-30. The only parts I used were an SPST toggle switch and a 1k resistor (see Fig. 1).

How It's Done

To add this modification to your AC-30, remove the circuit board from the case. Next, on top of the board, cut the trace between pins 10 and 11 of IC14 —it will be under the IC or its socket. I discovered that on the type of sockets I used (I always use sockets) the plastic housing could be pried off the pins and replaced after cutting the trace. What luck!

Now, on the bottom of the board connect a wire from IC14, pin 11 to an unused pin on the rear center board connector; next on the bottom, solder a 1k resistor, R1, from IC14, pin 11 to IC14, pin 16, which is +5 volts. Connect one side of the switch to the male pin that matches the vacant board connector you used. The other side of the switch connects to the pin that matches the rear center connector labeled LOCAL/$\overline{\text{REMOTE}}$ on the placement diagram.

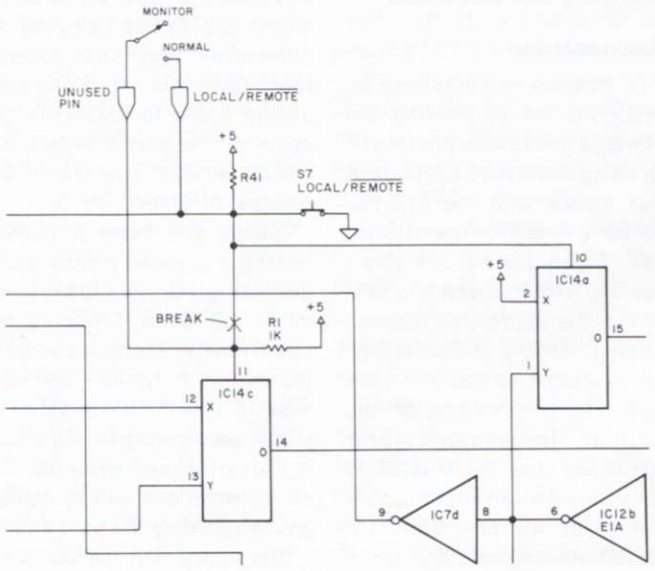
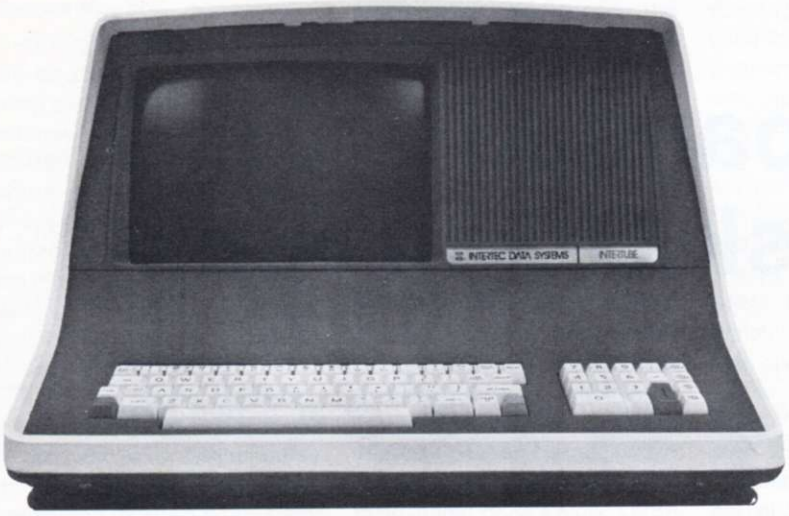


Fig. 1. Section of AC-30 cassette-tape-switching schematic showing modification.

I mounted the switch at the right rear of the cabinet, out of sight but still accessible. I have since considered moving the power switch to the back and the monitor switch to the front because sometimes I forget to put it in the normal position and wonder why the computer is not speaking to me.

When you want to monitor a tape you are loading, place the switch in its open position. This will change data selector IC14c from the computer to the cassette demodulator output. This causes the data to be sent to the computer and to the terminal. When the switch is in its normal position (closed), the AC-30 operates normally. This modification is especially useful when you're loading someone's tape other than your own for the first time without having to check its readability in the LOCAL mode first.

If the computer receives bad data during a normal load, it will send question marks to the terminal. Remember: With the switch in the monitor position, the terminal is looking at the AC-30, not the computer; but if there is bad data, the TVT should show it anyway. ■



The InterTube II Video Display Terminal is truly representative of the latest state-of-the-art advances in microprocessor technology. Its basic teletypewriter compatability combined with its numerous "smart" terminal features satisfy the universal requirement for a low-cost, high performance video terminal.

You get everything you need. An upper and lower case character set displayed on a sharp 8 X 10 dot matrix. A full 24 line by 80 character screen. A status line displayed in reverse video. A complete ASCII keyboard with an 18-key numeric pad.

You get full cursor addressing, automatic repeat of all keys and individual backspace and shiftlock keys. Plus, a graphics mode for easy design and display of all types of forms. And an RS-232 serial printer port.

And you get everything your operators need to make their jobs a pleasure. A hooded display that cuts glare and gives extra privacy. A wide bandwidth monitor for sharp images everywhere on the screen. Below-the-line character descenders to make reading easier. A programmable white-on-black or black-on-white display and a self-test mode for easy maintainability.

You get high powered text editing with such features as character and line insert/delete, full and/or partial block transmit, programmable end-of-line terminators, and protected fields. All standard! And all for a retail price you won't believe . . . only $995. Incredible!

INTERTEC'S

INTERTUBE II®

VIDEO DISPLAY TERMINAL

ATTENTION OEM's and DEALERS:

Your customers request InterTube terminals for one simple reason. They outperform the competition so well that it's foolish to consider any other terminal. Add to that InterTube's rugged design which insures you of the reliability that brings customers back. And modular design engineering that makes service a snap!

But best of all, the InterTube is readily available. Just a quick call and you'll have units in stock. Immediately! And our scheduled delivery program will help you keep them in stock.

Good margins, good service, good delivery. Simple? You bet it is! InterTube II dealerships and OEM agreements are now available in many areas. Contact us today and start selling from stock tomorrow!

INTERTEC DATA SYSTEMS® ✓121

Corporate Headquarters: 2300 Broad River Road • Columbia, South Carolina 29210 • 803 / 798-9100 • TWX: 810-666-2115

✓ Reader Service—see page 227

Expanded TRS-80 Disk Operations

Putting "system" programs on disk is the topic in this, the first of a two-part series.

Allan J. Domuret
7825 Willowcrest Way
Fair Oaks CA 95628

This article has two objectives. One is to explain how to put low-RAM machine-language programs on disk, a procedure that is not routinely supported by the Radio Shack disk operating system (DOS). Second, the techniques described herein will serve as a good introduction to machine-language programming for beginners.

The techniques make use of the Small System Software RSM2 or RSM2D machine-language utility program. RSM2 has many features, and only a few are described herein. If the user does not have RSM2 in his software library, this article can still be educational. It is suggested, however, that RSM2 or RSM2D is an invaluable utility for TRS-80 owners. RSM2D is the disk version of RSM2.

TRS-80 Disk Operations

The comparatively low price of the TRS-80 microcomputer, no doubt, contributes considerably to its popularity, but, as an old philosopher once said, "You get what you pay for." It seems that whenever a TRS-80 user updates or adds to his system, he encounters some kind of new frustration. What I refer to is a deficiency in Radio Shack's DOS, which precludes loading system programs such as the Electric Pencil, EDTASM or T-BUG from disk because these programs reside in the same memory area as does DOS.

To quote from the new Radio Shack "Disk Operating System" manual for TRSDOS Version 2.0: "Most Radio Shack system tapes designed for use with Level II TRS-80s will not work under Disk BASIC because of differences in RAM usage under Disk BASIC and Level II" (pp. 5-6). On the same page of the DOS manual is a comment that states, "Do not attempt to use Tapedisk to load tape files that load below hexadecimal address 54F4 (decimal 21748). Tapedisk uses this area." (Tapedisk is a utility program that allows the user to place system tapes on disk, provided that memory required by the system program does not overlap that required by Tapedisk or DOS.) In other words, it is not possible (normally) to put a machine-language program on disk if the program conflicts with DOS in low RAM.

This means that the inexperienced TRS-80 user cannot easily put system programs such as T-BUG, Electric Pencil or Editor/Assembler on disk with available routines such as Tapedisk. With a little skill and cunning, however, it is possible to load these low-RAM system programs from disk, thereby eliminating the need for much slower tape loading.

The technique involves loading the desired system program into upper, unused memory with RSM2. Then a short block-move subroutine (15 bytes) is permanently attached to the system program to automatically move the entire system program to its proper memory location for execution. Finally, the Tapedisk routine is used to put the program and its block-move subroutine onto disk.

Subsequently, when the program is loaded from disk, it is loaded into upper memory with an execution address that initiates the block move. The block move takes over and transfers the program to its proper memory location. When the block move is completed, a jump to the program's execution address completes the operation. The program will then run normally.

Moving the system program to lower RAM in this way overwrites DOS because DOS resides in the same memory location as the moved system program. But this is inconsequential because only the system program will be run—DOS will not be needed while the system program is running. For example, DOS and the Electric Pencil normally reside in the same memory area, but only one program will be used at a time. Wiping out DOS by overwriting it with the Electric Pencil does nothing to interfere with normal Pencil operations.

The "Save-to-Disk" technique described in this article will use the Electric Pencil as an example, but the reader, if he understands what is happening in this simple routine, can apply the same technique to other system programs such as T-BUG, Radio Shack's Editor/Assembler, and so on.

The recent release of a disk version of the Electric Pencil for the TRS-80 will not obviate the relevance of the procedures described in this article. The technique is still applicable to any system program, and, no doubt, there are those who may not be able to afford the new disk version of the Pencil.

Also, the ability to load the Editor/Assembler from disk eliminates much of the drudgery from machine-language programming. It only takes a moment for your machine-language program to crash, but it takes many frustrating minutes to reload the EDTASM and the source module for making repairs.

Part 2 of this article will explain how to put Peter Jenning's Microchess 1.5 on disk and also how to make backup copies on cassette tape. As most TRS-80 users have become aware, Microchess 1.5 cannot be duplicated on cassette tape by ordinary means. You paid good money, however, to purchase Microchess, and we all know the importance of having backups of valuable programs. It is also an educational experience to learn how to duplicate Microchess. I can only encourage TRS-80 users not to abuse this knowledge by making free copies for friends.

(Sorry, Mr. Jennings, but knowledge must be shared.)

RSM2 and RSM2D System Monitor

As I stated, RSM2 and RSM2D will be used as a tool to accomplish our purpose. The only difference between RSM2 and RSM2D is that the latter has some disk-related features. These disk features are not required for the techniques employed in this article, so they will not be referred to again. Subsequent references to RSM2 apply as well to RSM2D.

RSM2 has several tape-related features that enable the user to read and write system tapes and to read virtually any machine-language tape into memory (including Microchess). These tape features are necessary for structuring system programs that will eventually be saved on disk.

Before proceeding, I should mention that either the 32K or 48K version of RSM2 is preferable to working with the 16K version. If you must work with the 16K version because of limited memory availability, it will be necessary for you to determine your own memory-address requirements as described in this article.

Using the 32K or 48K version of RSM2 should present no problems because most TRS-80 users quickly learn that a 16K system with DOS is really not enough. Consequently, they usually move up to 32K or 48K of memory rather quickly. Those of you who will be using the 32K or 48K RSM2 can use the program address numbers exactly as provided in the following paragraphs.

The first step in creating our disk file is to determine the start, end and execution addresses of the selected system program. As was stated above, we will use the Electric Pencil for our example. To determine these addresses, load either the 32K or 48K version of RSM2 as you would any typical system program.

If you are still limited to a 16K TRS-80, it will be necessary to work in BASIC II rather than in Disk BASIC. The reason is that Disk BASIC occupies memory space that, when added to memory space used by RSM2, leaves an insufficient amount of memory for our operations on the Pencil program. In spite of limitations imposed by 16K systems, the techniques described herein will still work for loading Pencil from disk.

After RSM2 is operational, it would be a good idea to clean up all memory locations to avoid confusion later when working on the Pencil. Before loading Pencil into memory, zero out all memory locations from 4200 hex to a place in memory just short of running into RSM2. For a 16K TRS-80, RSM2 resides at 6C00 to 7EFF hex (see "Supplementary Information" provided with RSM2 documentation). The command for a 16K system would be: "Z 4200 6BFF." For the other RSM2 versions, zeroing memory from 4200 to, say, 8000 will be adequate for our purposes.

Now when we load in the Pencil program, there will be nothing else in lower RAM to confuse us as to which instructions belong to Pencil and what might otherwise be hanging around as leftovers from DOS or some other program that is no longer needed. The only memory locations that will not be zeroed will be the Pencil program, which we will load.

Prepare the cassette recorder and read in Pencil with the RSM2 "R 0" command. When the Pencil is loaded, RSM2 will display the file name—PENCIL—and it will also display the execution address. For Pencil, the execution address will be 4350 hex. Write down the execution address for later use.

By the way, in case you wondered, the Pencil execution address is physically located at the end of the cassette tape program, and this holds true for most system programs. This is how the TRS-80 knows where to go to execute a system program when you enter the slash (/) after normal tape loading, and this is how RSM2 is able to tell you what the execution address is.

If you want to check this out, read the Pencil into memory with the RSM2 "R" command (in contrast to the R 0 command). The R command will load in the program and will display the execution address at the end of the program. The R command is used for reading in non-system programs, but will read in system programs with all coding characters preserved in memory. The execution address for a system program will follow the termination code "78" at the end of the tape.

After loading Pencil with the RSM2 R 0 (not R) command, Pencil should be in its normal memory location, but RSM2 will still be in command of the machine. To determine Pencil's start and end addresses, use either the RSM2 ASCII or symbolic dump commands. The ASCII dump is faster for locating the Pencil in memory, but the symbolic dump is more precise in determining exact start and end addresses.

To get an ASCII dump, answer RSM2's "COMMAND?" prompt with: "A 4300." This will give a scrolling ASCII display of everything in memory starting from memory location 4300 hex. Stop the scroll as desired with the space bar. Since we initially zeroed all memory locations before starting, and since RSM2 resides in higher memory, we can be certain that the first nonzero data encountered during the ASCII or symbolic dump belongs to the Pencil that we loaded in with RSM2.

Since the ASCII dump makes it difficult to precisely read program addresses, the exact start address for the Pencil can be determined with the RSM2 symbolic dump. Start the symbolic dump at, say, 4300 hex by typing in: "S 4300." Notice that adjacent to the displayed memory locations between 4300 and 434F are zeros (00) and NOPs. This is because we zeroed all memory before loading in the Pencil. When the dump reaches 4350 hex, the start address for the Pencil, you will see:

```
4350: C3 DA 52        JP        52DA
```

followed by the rest of the program in Zilog mnemonics.

The first column (4350) is the hex memory location of the first instruction in the program; the second column group (C3 DA 52) represents the hex op codes, which tell the computer what to do; and the next column group (JP 52DA) represents the symbolic code to JUMP to memory location 52DA hex.

To find the end address, just let the memory scroll continue until a long string of zeros is again encountered. By now, the importance of zeroing all unused memory as we did at the start should be obvious. The last instruction in the Pencil program should read:

```
5365: 29              ADD       HL,HL
```

The end address is 5365, and the 29, rather than representing a machine-language op code, is actually an ASCII character for ")". An "ADD HL,HL" symbolic instruction is displayed because the RSM2 disassembler interprets the 29 op code as an "ADD HL,HL" instruction. RSM2 really has no way of telling if the 29 is an ASCII character or an op code without actually running the Pencil program.

This is a common fault for most disassemblers, and the user should always be aware of this problem so that ASCII characters are not inadvertently interpreted as op codes. ASCII characters can be found easily with RSM2 by dumping memory with the ASCII dump command and finding the English language text in the program. For Pencil, this English language text is found in upper memory.

At this point, I must add that Pencil is recorded on the cassette tape with ten trailing zeros after the last code, 29. I learned this by loading Pencil into memory with the RSM2 R command, in contrast to the R 0 command. The R command will read virtually any tape, whereas the R 0 command is for reading only system tapes.

If you care to experiment, read in Pencil using the RSM2 R command, and at the end of the program you will find the ten trailing zeros, followed by a few other characters, including the '78' termination code and, at the end, the execution address as I mentioned above. Little peculiarities like these must be found when manipulating programs at the machine-language level. At